Coding theory: the unit-derived methodology

Ted Hurley*

Department of Mathematics,
National Universiy of Ireland Galway,
Galway, Ireland
Email: Ted.Hurley@NuiGalway.ie
*Corresponding author

Donny Hurley

Department of Computing and Electronic Engineering,
Institute of Technology Sligo,
Ireland
Email: hurleyd@yahoo.com

Abstract: The unit-derived method in coding theory is shown to be a unique optimal scheme for constructing and analysing codes. In many cases, efficient and practical decoding methods are produced. Codes with efficient decoding algorithms at maximal distances possible are derived from unit schemes. In particular unit-derived codes from Vandermonde or Fourier matrices are particularly commendable giving rise to mds codes of varying rates with practical and efficient decoding algorithms. For a given rate and given error correction capability, explicit codes with efficient error correcting algorithms are designed to these specifications. An explicit constructive proof with an efficient decoding algorithm is given for Shannon's theorem. For a given finite field, codes are constructed which are 'optimal' for this field.

Keywords: code; unit-derived schemes; decoding.

Reference to this paper should be made as follows: Hurley, T. and Hurley, D. (2018) 'Coding theory: the unit-derived methodology', *Int. J. Information and Coding Theory*, Vol. 5, No. 1, pp.55–80.

Biographical notes: Ted Hurley is a Professor of Mathematics at the National University of Ireland Galway. He previously held positions at Imperial College (University of London), the University of Sheffield and at the University College Dublin. He held visiting positions at the Australian National University (Canberra) and at the University of Alberta (Edmonton). His research interests involve combinatorial group theory, group rings and algebra in communications including coding theory, cryptography and compressed sensing.

Donny Hurley is a Lecturer in the Department of Computing and Electronic Engineering at the Institute of Technology Sligo. He previously worked at Maynooth University. His research interests include aspects of mathematics in computer science.

1 Introduction and background

Error-correcting codes are used extensively in communications' applications including digital video, radio, mobile communication, satellite/space communications and other systems.

Here the unit-derived method is exploited to design maximum distance separable codes with efficient decoding algorithms. For a given rate and a given error-correcting capability, codes with efficient decoding algorithms are designed to these specifications and are shown algebraically to have the required properties. This is used to give explicit codes with efficient decoding algorithms to prove Shannon's theorem.

Section 1.2 gives further details on content and results. Samples demonstrating the extent of the constructions are given. Some well-known codes in practical use are shown to be special cases; better-performing ones can be designed from the general techniques.

Background on coding theory may be found in Blahut (2003), McEliece (2002) and others. Most of the algebraic background may be found in Blahut (2003) and further background on algebra and coding theory is developed or referenced as required.

Now (n, r, d) denotes a code of length n, dimension r and (minimum) distance d. The rate of the code is $\frac{r}{n}$. The code (n, r, d) can correct $t = \lfloor \frac{d-1}{2} \rfloor$ errors and this is the error-correction capability of the code. The code is called a *maximum distance separable* (mds) code if it of the form $(n, r, n - r + 1)$, that is, if it attains the maximum distance allowable for a given length and dimension.

$GF(q)$ denotes the finite field of q elements where $q = p^s$ is a power of a prime p. The units of $GF(q)$ are the non-zero elements of $GF(q)$ and these units form a cyclic group generated by a primitive $(q - 1)$th root of unity in $GF(q)$. For a prime p, $GF(p) = \mathbb{Z}_p$, the integers modulo p.

1.1 Unit-derived codes

In Hurley and Hurley (2009), and also in Hurley and Hurley (2007, 2010a), methods are developed for constructing *unit-derived codes*; these methods are fundamental. The unit-derived schemes may be described briefly as follows. Let $R_{n\times n}$ denote the ring of $n \times n$ matrices with entries from R, a ring with identity, often a field but not restricted to such. Suppose $UV = I_{n\times n}$ in $R_{n\times n}$. Taking any r rows of U as a generator matrix defines an (n, r) code and a check matrix is obtained by deleting the corresponding columns of V. Further details may be found in expanded book chapter form in Hurley and Hurley (2010a).

Now R can be any ring with identity and it has been useful to consider cases other than fields; cases where R is taken as a polynomial ring, a group ring or as a matrix ring has been useful in constructing different types of codes such as LDPC codes or Convolutional codes, Hurley and Hurley (2010b), Hurley et al. (2010) and Hurley (2009, 2016a).

From the unit scheme $UV = I$, the first r rows in particular of U may be taken as the generator matrix of a code and then the last $(n - r)$ columns of V give a check matrix for this code. Thus if $UV = I_n$ and $U = \begin{pmatrix} A \\ B \end{pmatrix}$ for an $r \times n$ matrix A and an $(n - r) \times n$ matrix B and $V = (C, D)$ for an $n \times r$ matrix C and an $n \times (n - r)$ matrix D, this gives $UV = \begin{pmatrix} A \\ B \end{pmatrix} (C, D) = I_n$ from which $\begin{pmatrix} AC & AD \\ BC & BD \end{pmatrix} = I_n$.

Thus $AD = 0_{r\times(n-r)}$ and D^{T} is a check matrix for the (n, r) code with generator matrix A. Note also that $AC = I_{r\times r}$, the identity $r \times r$ matrix, and this will be useful later.

Any linear code is equivalent to a unit-derived code but there may not be any advantage in using the equivalence.

Using the unit-derived method has many advantages. Unit-derived codes are in general not ideals; cyclic and some other such codes are ideals in group rings. Many different codes of various rates and with predetermined properties may be constructed from a single unit scheme. Properties of the units may be used to derive codes of particular types and/or with particular properties. From the set-up, more information on the code $\mathcal{C}$ is available than just its generator and check matrix. Here also efficient decoding methods for certain unit-derived codes are established.

In the unit scheme as above, $\begin{pmatrix} A \\ B \end{pmatrix}(C, D) = \begin{pmatrix} AC & AD \\ BC & BD \end{pmatrix}$, A is taken as the generator of a code. If αA is a codeword then $\alpha A * C = \alpha$. Hence the originally transmitted vector α may be obtained by multiplying on the right by C once the errors have been eliminated by an error-correcting method.

1.2 Layout and summary

General theorems, Theorems 3.1, 3.2, required for the constructions and decoding methods are stated in Section 3; these are proved later in Section 5.

Section 4 presents examples as an introduction to, and illustration of, the general techniques resulting from Theorems 3.1 and 3.2. These examples have interest in themselves, have full distances and implementable practical decoding algorithms. The examples are far from exhaustive and could be considered as prototypes for many others.

An illustrative example in Section 4.2 demonstrates the decoding method which is later derived in general in Section 6.

Section 5 introduces the general method and derives background results from which the properties of the unit-derived codes may be deduced and from which the decoding algorithms are created. Results on Vandermonde/Fourier matrices are developed; unit-derived codes from these are particularly commendable with schemes for deriving maximum distance separable codes with practical decoding algorithms. Section 6 derives the general decoding algorithms.

Section 7 describes the general method of constructing codes with required rate and required error-correcting capability; Section 7.1, gives examples of such required yield constructions. Section 8 uses the methods to derive an explicit proof of Shannon's theorem with an efficient decoding algorithm.

Section 7.4 notes 'optimal' codes for a particular finite field.

The use of the unit-derived method for defining and analysing particular types of codes such as LDPC (Low density parity check) codes, Convolutional Codes and others is discussed in Section 2. Section 2.4 suggests using the codes for cryptographic schemes.

2 Construction of special types[1]

Low density parity check (LDPC) codes and convolutional codes attract much attention. Unit schemes are and have been used to generate such codes by relating the prescribed properties to properties of the units from which they are derived.

2.1 Low density

A low density parity check (LDPC) code is a linear code where the check matrix has *low density* which means that each row and column has only a small number of non-zero entries compared to the size of the matrix.

An LDPC code may be obtained from a unit scheme $UV = I_n$. To do this, we must be able to choose columns of V to form a (check) matrix which has low density compared to its size. The columns of V chosen to decide the rows of U to be used in generating the code. See Hurley and Hurley (2010b) and Hurley et al. (2010) for further details.

One way to ensure that any choice of rows will be an LDPC code is to ensure that V itself has low density in all its rows and columns. Indeed from such a unit system with V of low density many (different) LDPC codes can be generated. It is also possible to find in general such V of low density so that the resulting LDPC codes have no short cycles Hurley et al. (2010); LDPC codes with no short cycles in the check matrix are known to perform well.

It may be shown that an LDPC code is equivalent to one derived from a unit scheme.

This method has been used successfully in Hurley et al. (2010) to generate large length LDPC codes with excellent performances.

2.2 Convolutional codes

The unit-derived method may be used to describe, define and study properties of Convolutional Codes, see Hurley and Hurley (2010b), Hurley (2009) ; here the unit schemes are over certain rings other than fields, such as polynomial rings or group rings. The reference Hurley and Hurley (2010b) in book chapter form is particularly written as an introduction to these methods.

The constructions in Hurley (2016a) may be considered as unit-derived convolutional code construction schemes which have parallels to the (linear) block code unit-derived schemes developed here.

2.3 Using group rings

Using the embedding of a group rings into a group of matrices, Hurley (2016b), allows the construction of self-dual, dual-containing, quantum codes, Hurley (2007), and other types from units in group rings. Cyclic codes are ideals in the group ring of the cyclic code. Unit-derived codes, in general, are not ideals.

2.4 McEliece type encryption

The codes that are or can be constructed from the unit-derived codes developed here can have a large length, have good error capability and good decoding capability and are thus suitable candidates for McEliece type encryption McEliece (1978). The problem with low rate data can be eliminated. Permutation of the rows and different selections may be used. This should be compared with the cryptographic schemes of Hurley (2014).

3 Main general results

Statements of the results from which the general constructions and decoding methods are derived are given in this section. The proofs of these follow from work in Sections 5 and 6.

Recall that an mds, maximum distance separable, code is one of the form $(n, r, n - r + 1)$ in which the maximum possible distance is obtained for a given length and dimension, see Blahut (2003) for details.

Theorem 3.1: *Let $V = V(x_1, x_2, \ldots, x_n)$ be a Vandermonde $n \times n$ matrix over a field $\mathbb{F}$ with distinct and non-zero x_i. Let $\mathcal{C}$ be the unit-derived code obtained by choosing in order r rows of V in arithmetic sequence with difference k. If $(x_i x_j^{-1})$ is not a kth root of unity for $i \neq j$ then $\mathcal{C}$ is an $(n, r, n - r + 1)$ mds code over $\mathbb{F}$. In particular the result holds for consecutive rows as then $k = 1$ and $x_i \neq x_j$ for $i \neq j$.*

For Fourier matrices the following theorem is obtained:

Theorem 3.2: *(i) Let F_n be a Fourier $n \times n$ matrix over a field $\mathbb{F}$. Let $\mathcal{C}$ be the unit-derived code obtained by choosing in order r rows of V in arithmetic sequence with arithmetic difference k and $\gcd(n, k) = 1$. Then $\mathcal{C}$ is an mds $(n, r, n - r + 1)$. In particular, this is true when $k = 1$ that is, when the r rows are chosen in succession.*

(ii) Let $\mathcal{C}$ be as in part (i). Then there exist efficient encoding and decoding algorithms for $\mathcal{C}$.

The decoding methods are based on the decoding methods used in Hurley (2017) in connection compressed sensing by solving underdetermined systems using error-correcting codes. These decoding methods themselves are based on the error-correcting methods due to Pellikaan Pellikaan (1992) which is a method of finding error-correcting pairs.

The complexity of encoding and decoding can be $\max\{O(n \log n), O(t^2)\}$ where $t = \lfloor \frac{n-r}{2} \rfloor$, that is where t is the error-correcting capability of the code. The complexity is discussed in Section 9.

4 Initial cases

Initial cases are presented as an introduction to, and illustration of, the general techniques.

The examples have interest in themselves and have practical decoding algorithms. They also serve as prototypes as to how general and longer length mds codes with efficient decoding algorithms may be constructed using the unit-derived method with Vandermonde/Fourier matrices. For the proofs that the codes constructed satisfy the mds and other properties, the reader is referred to Section 5 and for the decoding algorithms, the reader should consult Section 6.

The reader might appreciate for comparison the mds codes (Section 4.4) of types $(255, 253, 3), (255, 251, 5), ..., (255, 155, 101), ...,$ or in general of type $(255, r, 256 - r)$, constructed over $GF(2^8)$ together with decoding algorithms. The methods may be extended to form mds codes over $GF(2^s)$ with decoding algorithms. It is shown that codes of the form $(256, r, 257 - r)$ may be generated over the prime field $GF(257)$ with decoding algorithms and these perform better.

4.1 To err is ...

If a code is required to correct one error it must have distance ≥ 3. If the length is also ≤ 3 then the code is equivalent to a repetition code, one of the form $(3, 1, 3)$.

For a code of length 4 to be 1-error correcting, and not a repetition code, it must be a $(4, 2, 3)$ mds code. Look at unit-derived codes from Fourier 4×4 matrices for such. No 4×4 Fourier matrix exists in characteristic 2 as $2|4$. Consider characteristic 3. Now $3^2 - 1 = 8$ so there exists an element of order 8 in $GF(3^2)$ and thus an element of order 4 exists in $GF(3^2)$. To construct $GF(3^2)$ use a primitive polynomial of degree 2 over $\mathbb{Z}_3 = GF(3)$ such as $x^2 + x + 2$. Then x has order 8 and $x^2 = \omega$ has order 4. Now form the 4×4 Fourier matrix F_4 over $GF(3^2)$ with ω as the primitive 4th root of 1.

By general theory, the first two rows or any two rows in succession of a Fourier F_4 matrix gives a generator matrix of a $(4, 2, 3)$ code. The rate of these codes is $\frac{2}{4} = \frac{1}{2}$.

Row 4 followed by row 1 also works but note that row 1 with row 3 will not give an mds code. Why?

The order of $GF(5)\backslash 0$ is 4. Then it is required to find an element of order 4 in $GF(5)$ and it is easily checked that 2 has order 4 modulo 5 as 3. Now form the Fourier 4×4 matrix over $GF(5)$ using $2 \mod 5$ as the primitive element: $F_4 = \begin{pmatrix} 1\,1\,1\,1 \\ 1\,2\,4\,3 \\ 1\,4\,1\,4 \\ 1\,3\,4\,2 \end{pmatrix}$. If the matrix is over $GF(5)$, the calculations can all be done with modulo 5 arithmetic.

A length 5 code could also correct 1 error if it is of the form $(5, 3, 3)$. The rate here is $\frac{3}{5}$. What is required is a Vandermonde or Fourier matrix of size 5×5 over a field. Such can be constructed in $GF(2^4), GF(3^4)$, ... but not in characteristic 5 of course.

For a length 6 code, it is required to construct a Vandermonde or Fourier 6×6 matrix and extract codes from the rows using the unit-derived method. A $(6, 2, 5)$ code can correct 2 errors but the rate is small. Consider constructing $(6, 4, 3)$ codes with 1-error correcting capability and rate $\frac{2}{3}$. $GF(7)$ has elements of order 6 such as 3 or 5 and these can be used to construct a Fourier 6×6 matrix over $GF(7) = \mathbb{Z}_7$. Taking the first four rows or any four rows in succession will generate a $(6, 4, 3)$ code over $GF(7)$.

All the small length codes mentioned here and below may be constructed directly using, for example, a package such as GAP, containing the coding sub-package GUAVA, reference et al. (20).

4.2 *Worked example of the decoding algorithm*

In Section 5 decoding algorithms are derived. Here an example of the workings of the decoding algorithms developed later is given.

Let $\mathbb{F} = GF(29)$. A generator of $\{\mathbb{F}\backslash 0\}$ has order 28. We are interested in a Fourier 7×7 matrix over $\mathbb{F}$. An element of order 7 is easily obtained in $\mathbb{F}$ and indeed $7^7 \equiv 1 \mod 29$.

Consider then the unitary scheme:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 \\ 1 & \omega^2 & \omega^4 & \omega^6 & \omega & \omega^3 & \omega^5 \\ 1 & \omega^3 & \omega^6 & \omega^2 & \omega^5 & \omega & \omega^4 \\ 1 & \omega^4 & \omega & \omega^5 & \omega^2 & \omega^6 & \omega^3 \\ 1 & \omega^5 & \omega^3 & \omega & \omega^6 & \omega^4 & \omega^2 \\ 1 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega \\ 1 & \omega^5 & \omega^3 & \omega & \omega^6 & \omega^4 & \omega^2 \\ 1 & \omega^4 & \omega & \omega^5 & \omega^2 & \omega^6 & \omega^3 \\ 1 & \omega^3 & \omega^6 & \omega^2 & \omega^5 & \omega & \omega^4 \\ 1 & \omega^2 & \omega^4 & \omega^6 & \omega & \omega^3 & \omega^5 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 \end{pmatrix} = 7I,$$

where ω is a primitive 7th root of unity. Here we may take $\omega = 7 \mod 29$ and powers of 7 are evaluated $\mod 29$. Other values for ω are possible and what is required is an element of order 7 modulo 29. [2] Let the first matrix above be denoted by P and the second by Q. Thus $PQ = 7 * I$ which is the unit scheme $P\{\frac{1}{7}Q\} = I$. Now choose r rows of P to form a matrix which generates a $(7, r)$ code and a check matrix for this code is obtained from Q by eliminating the columns corresponding to the chosen rows of P; in theory the check matrix is from $1/7 * Q$ but if H is a check matrix then so is $7 * H$.

From P then $(7, 3, 5)$ and $(7, 5, 3)$ codes may be obtained by taking, in particular, the first 3 rows or 5 rows of P or indeed by taking the required number of rows consecutively from P. The general theory which verifies this, including the distances obtained, is given in Section 5 below.

A $(7, 5, 3)$ code is 1-error correcting. Take the first 5 rows of P as the generator matrix A and then the last two columns, D, of V is the check matrix. A codeword is αA for a 1×5 vector α. Suppose $\alpha A + \epsilon$ is received where ϵ is the error and has just one non-zero entry. Applying D to $\alpha A + \epsilon$ gives ϵD. Now ϵD is a multiple of a row of D as ϵ has only one non-zero entry, and this uniquely defines the row and its multiple. Thus the error ϵ may be eliminated. When the error has been eliminated, then $\alpha A * C = 7 * \alpha$ decodes the word where C denotes the first 5 columns of Q.

This decoding method of identifying the multiple of the row of the check matrix works whenever just 1-error needs correcting.

A 2-error correcting code $(7, 3, 5)$ is obtained from this unit scheme by taking any three rows of P as a generator matrix. The code may be corrected as follows; the details of the algorithm may be found in Hurley (2017) which was derived from the error-correcting methods of Pellikaan (1992). The algorithm utilises error-correcting pairs which are shown to exist for these codes.

Suppose the first 3 rows are the generator matrix of a code $\mathcal{C}$. Then the last 4 columns of Q constitute a check matrix. Let these columns be denoted by $\{{E_4}^{\mathrm{T}}, {E_3}^{\mathrm{T}}, {E_2}^{\mathrm{T}}, {E_1}^{\mathrm{T}}\}$ in order. Then $\mathcal{C}^{\mathrm{T}}$ is generated by these columns, written as rows. The first three rows of P are $\{E_0, E_1, E_2\}$ where E_0 consists of all 1^s.

Now by Hurley and Hurley (2014) and Pellikaan (1992) an error-correcting pair for $\mathcal{C}$ is as follows:

$U = \langle E_1, E_2, E_3\rangle, V = \langle E_0, E_1\rangle$ are error correcting pairs for $\mathcal{C}$.

Let αA be the codeword but when transmitted an error is introduced and the word received is $\alpha A + w$. Note w is a 1×7 vector. Apply the check matrix which has columns $\{{E_4}^{\mathrm{T}}, {E_3}^{\mathrm{T}}, {E_2}^{\mathrm{T}}, {E_1}^{\mathrm{T}}\}$ and then $< w, E_i >= w{E_i}^{\mathrm{T}} = E_i w^{\mathrm{T}}$ are known for $i = 1, 2, 3, 4$ where $<, >$ denotes inner product. Let $< w, E_1 >= \alpha_1, < w, E_2 >= \alpha_2, < w, E_3 >= \alpha_3, < w, E_4 >= \alpha_4$. The algorithm then is:

- Find an element x^{T} in the kernel of $\begin{pmatrix} \alpha_1 \, \alpha_2 \, \alpha_3 \\ \alpha_2 \, \alpha_3 \, \alpha_4 \end{pmatrix}$. Any non-zero element of the kernel will do.
- Form $\underline{a} = (E_1, E_2, E_3)x^{\mathrm{T}}$.
- Find the locations of the zero coefficients of $\underline{a}$. Say these are at j_1, j_2 for $1 \le j_1, j_2 \le 7$.

- Solve $\begin{pmatrix} E_{1,j_1} & E_{1,j_2} \\ E_{2,j_1} & E_{2,j_2} \\ E_{3,j_1} & E_{3,j_2} \\ E_{4,j_1} & E_{4,j_2} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix}$. Here $E_{k,l}$ denotes the lth entry of E_k.
- w is then x_1, located at j_1, and x_2, located at j_2, and zeros elsewhere.

Suppose now that $\omega = 7 \in GF(29)$ is taken as the 7th root of unity of the Fourier matrix and the α_i are found to be: $\alpha_1 = 18, \alpha_2 = 15, \alpha_3 = 4, \alpha_4 = 12$. Then

- An element in $\ker \begin{pmatrix} 18 & 15 & 4 \\ 15 & 4 & 12 \end{pmatrix}$ is $x^{\mathrm{T}} = (23, 5, 1)^{\mathrm{T}}$
- $\underline{a} = (E_1, E_2, E_3)x^{\mathrm{T}} = (0, 24, 20, 1, 0, 2, 11)$. This has zeros at positions $j_1 = 1, j_2 = 5$.
- Solve $\begin{pmatrix} E_{1,j_1} & E_{1,j_2} \\ E_{2,j_1} & E_{2,j_2} \\ E_{3,j_1} & E_{3,j_2} \\ E_{4,j_1} & E_{4,j_2} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix}$ is then solve $\begin{pmatrix} 1 & 23 \\ 1 & 7 \\ 1 & 16 \\ 1 & 20 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 18 \\ 15 \\ 4 \\ 12 \end{pmatrix}$. This has solution $x_1 = 1, x_2 = 2$.
- Then the error is x_1 located at $j_1 = 1$ position and x_2 located at position $j_2 = 5$ giving the error vector $w = (1, 0, 0, 0, 2, 0, 0)$.

The calculations, in this case, are all done in $\mathbb{Z}_{29} = GF(29)$.

4.3 Further samples

4.3.1 11×11 *cases*

Suppose a Vandermonde or Fourier 11×11 matrix F_{11} over a field $\mathbb{F}$ has been found. Now choose rows consecutively[3] to construct codes, and error-correcting pairs exist for these codes. In Section 5 below it is shown that such codes from F_{11} are mds, maximal distance separable codes and decoding methods are derived in Section 6.

Thus $(11, 3, 9)$ codes which have 4-error correcting capability, $(11, 5, 7)$ which have 3-error capability, $(11, 7, 5)$ which have 2-error correcting capability, and $(11, 9, 3)$ which have 1-error capability ability are obtained. The decoding algorithms reduce to finding t-error correcting pairs.

An example of such a field which has an easily workable 11th of unity is $GF(23)$. The group of non-zero elements in $GF(23)$ is of order 22 and is cyclic so elements of order 11 exist. In fact, $2 \mod 23$ or $3 \mod 23$ have order 11 in $GF(23)$ and either of these may be used as a primitive 11th root of unity in forming F_{11}. The calculations, in this case, are the arithmetic modulo 23.

Let F_{11} denote the Fourier matrix in $GF(23)$ with $\omega = 2 \mod 23$ as the primitive 11th root of unity. Take consecutive rows or else select rows in arithmetic sequence of their order. An efficient decoding algorithm using error correcting pairs exists for these codes is given generally in Section 6; the algorithm is derived from Hurley (2017).

Notice that 11 divides $2^{10} - 1$ so the Fourier matrix of size 11×11 can also be constructed over $GF(2^{10})$. However, this field is large and calculations may be more difficult. But see Section 4.4 below for discussion of characteristic 2 cases which have other advantages.

Note that 11 divides $3^5 - 1$ so the field $GF(3^5)$ could also be used.

4.3.2 13×13 *cases*

For 13×13 Fourier matrices, there are a number of possibilities. To work in modular arithmetic take $\mathbb{F} = GF(53)$ as 13 divides $(53 - 1) = 52$, and then there exists primitive 13th of unity. In fact $10^{13} \equiv 1 \mod 53$ so $10 \mod 53$ may be used as the primitive 13th root of unity in $GF(53)$ in forming the Fourier 13×13 matrix.

In $GF(3^3)$ also there exists a 13th root of unity as $3^3 - 1 = 26 = 2 * 13$. So indeed the square of the generator of the non-zero elements of $GF(3^3)$ is a primitive 13th[4] root of unity. Use an irreducible primitive polynomial of degree 3 in $\mathbb{Z}_3 = GF(3)$ with which the calculations may be made in $GF(3^3)$.

4.4 Characteristic 2 cases

Characteristic 2 cases are always interesting and this is indeed the case with these unit-derived codes from Vandermonde/Fourier matrices.

Codes over $GF(2^s)$ may be transmitted as binary signals. The code symbols are within $GF(2^s)$. If each code symbol is represented by an s-tuple over $GF(2)$, then the code can be transmitted using binary signalling. In decoding, every s received bits are grouped into a received signal over $GF(2^s)$.

- As $2^2 - 1 = 3$ so 3×3 Fourier matrices over $GF(2^2)$ can be obtained and mds codes may be derived from this. These, however, are equivalent to repetition codes $(3, 1, 3)$ or to codes of the form $(3, 2, 2)$ which do not have error-correcting capabilities.
- $2^3 - 1 = 7$ gives a Fourier 7×7 matrix over $GF(2^3)$. Thus codes $(7, 3, 5)$ which are 2-error correcting and codes $(7, 5, 3)$ which are 1-error correcting may be formed over $GF(2^3)$.
- $2^4 - 1 = 15$ and so $(15, 13, 3), (15, 11, 5), (15, 9, 7), (15, 7, 9)$ codes can be formed by this method over $GF(2^4)$.
- $2^5 - 1 = 31$, which is prime, enables $(31, 29, 3), (31, 27, 5), (31, 25, 7), (31, 23, 9), ...$ codes to be formed over $GF(2^5)$. If rate about $3/4$ is required then take $(31, 23, 9)$ which is 4-error correcting.
- $2^6 - 1 = 63$. Codes of form $(63, r, 64 - r)$ may be formed with efficient error-correcting algorithms.
- $2^7 - 1 = 127$. Fourier 127×127 matrices may be formed over $GF(2^7)$. Note that 127 is prime, in fact, a Mersenne prime, and Fourier matrices of length a Mersenne prime are interesting. Here mds codes of the form $(127, 125, 3), (127, 123, 5), ..., (127, 87, 41), ...$, may be formed using unit-derived codes from this Fourier matrix over $GF(2^7)$. Note for example that $(127, 97, 31)$ has rate $\frac{97}{127} > \frac{3}{4}$ and can correct 15 errors.
  Use a prime field? From the prime field $GF(127)$ a Fourier 126×126 matrix may be formed with elements from $GF(127) = \mathbb{Z}_{127}$ and unit-derived codes may be constructed from this; the algebra then is $\mod 127$.
- Now $2^8 - 1 = 255$ and this is an interesting case as mds codes over $GF(2^8)$ are in practical use. The Reed-Solomon (see for example Blahut (2003)), $(255, 239, 17)$

code over $GF(2^8)$ is used extensively in data-storage systems, hard-disk drives and optical communications; the Reed-Solomon $(255, 223, 33)$ code over $GF(2^8)$ is or was the NASA standard for deep-space and satellite communications.
Form the Fourier 255×255 matrix using a primitive 255th root of unity in $GF(2^8)$. A primitive polynomial of degree 8 over $\mathbb{Z}_2 = GF(2)$ would be useful here; lists of these are known and one such is $x^8 + x^4 + x^3 + x^2 + 1$. By taking unit-derived codes from this Fourier matrix one readily gets
$(255, 253, 3), (255, 251, 5), ..., (255, 239, 17), ..., (255, 223, 23), ..., (255, 155, 101), ...$ codes. So, for example, the code $(255, 155, 101)$ can correct 50 errors. Practical error-correcting algorithms for these are given within the general form of Section 5.
A better way perhaps of constructing these types of codes is to consider the prime 257 and then the field $GF(257)$. The order of the units of $GF(257)$ is 256 and then construct the Fourier 256×256 matrix over $GF(257)$ using a primitive 256th root of unity. Now the order of $3 \mod 257$ is 256 so indeed $3 \mod 257$ could be used as this primitive root of unity in forming the Fourier 256×256 matrix over $GF(257)$. Other primitive generators could be used such as 5 as the order of $5 \mod 257$ is also 256. Note here also that the arithmetic is modular arithmetic in $\mathbb{Z}_{257} = GF(257)$. For example codes of form $(256, 222, 35)$ with efficient decoding algorithm which can correct 17 errors may be formed over $GF(257)$; indeed codes of the form $(256, r, 257 - r)$ may be formed over $GF(257)$ with efficient decoding algorithms for $1 \leq r \leq n$.

- Clearly, also one can go much further and work with $GF(2^s)$ for $s > 8$.

4.5 Using special fields

Suppose we require that the Fourier matrix, from which the unit-derived codes are generated, be of size $p \times p$ for a prime p.

4.5.1 Mersenne and repunit primes

Fields of characteristic 2 were considered in Section 4.4.

Suppose the generator of the non-zero elements of $GF(2^s)$ is of order a prime p and form the Fourier $p \times p$ matrix using this generator as the pth root of unity. This gives a $p \times p$ matrix over $GF(2^s)$ from which unit-derived mds codes may be generated; these have nice properties. For example when rows are selected in arithmetic sequence k then always $\gcd(n, k) = 1$ and the resulting codes have efficient decoding algorithms.

Saying the non-zero elements of $GF(2^s)$ have ordered a prime is simply saying that $2^s - 1$ is a Mersenne prime. The first Mersenne primes are $3, 7, 31, 127, ..$, but it is unknown if there are an infinite number of these.

The fields $GF(2^5)$, $GF(2^7)$ with $2^5 - 1 = 31$ and $2^7 - 1 = 127$ were given as examples in Section 4.4.

All these have efficient error-correcting algorithms as explained in Section 6.

One can also consider *repunit base p primes*. Now q is a repunit base p prime if q is a prime and $p^s - 1 = (p - 1)q$ for some s. Repunit base 2 primes are the Mersenne primes. Using repunit base p prime q with $p^s - 1 = (p - 1)q$ leads to considering $q \times q$ Fourier matrices over $GF(p^s)$. Details are omitted.

4.5.2 Germain primes

It is often useful to have a prime size Fourier matrix in as small a field as possible. If this field is also a prime field, then this is even better as the calculations are then modular arithmetic over the prime field. Thus we are lead to consider Germain primes. Now p is a *Germain prime* if $2p + 1$ is also a prime.

Consider the field $GF(2p + 1)$ where p is also a prime. A generator ω of the non-zero elements of $GF(2p + 1)$ has order $2p$ and thus $\alpha = \omega^2$ has order p. Now form the Fourier $p \times p$ matrix over $GF(2p + 1)$ using α as a primitive pth root of unity. Codes are then formed from the rows of this Fourier matrix and these are mds codes with efficient decoding algorithms. As the codes are over $GF(2p + 1)$ the arithmetic is modular arithmetic over $\mathbb{Z}_{2p+1}$.

The first Germain primes are $2, 3, 5, 11, 23, 29, 41, \ldots$.

For example $p = 29$ gives $2 * p + 1 = 59$ and form a Fourier 29×29 matrix over $GF(59)$ using the square of any generator of the non-zero elements of $GF(59)$. The order of 2 $\mod 59$ is 58 so the order $4 \mod 59$ is 29; however, the order of 3 $\mod 59$ is also 29 and this is preferable. Thus take $\omega = 3 \mod 59$ and form the Fourier 29×29 matrix over $GF(59)$ using this ω as the primitive 29th root of 1.

5 General enabling results

In Hurley (2017) conditions are given to ensure that subdeterminants of Vandermonde matrices are non-zero. Fourier matrices are special types of Vandermonde matrices. Such conditions can be applied to generate codes from units with maximum possible distance and further it is shown that practical decoding algorithms for these codes exist.

Of particular relevance in Hurley (2017) is Section 6, noting Proposition 6.1 and its corollaries.

5.1 Determinants of submatrices

The Vandermonde matrix $V = V(x_1, x_2, \ldots, x_n)$ is defined by

$$V = V(x_1, x_2, \ldots, x_n) = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_n \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \ldots & x_n^{n-1} \end{pmatrix}.$$

It is assumed that entries of a Vandermonde matrix here are over a field and not necessarily over the real or complex numbers. It is well-known that the determinant of V is non-zero if and only if the x_i are distinct; in fact $\det V = \prod_{i<j}(x_i - x_j)$.

Assume, in addition, from now on that *all entries of a Vandermonde matrix used here are non-zero*.

The following Proposition and its corollaries are taken from Hurley (2017). The proofs are included again here for completeness and for their importance.

Proposition 5.1: *Let $V = V(x_1, x_2, \ldots, x_n)$ be a Vandermonde matrix with rows and columns numbered $\{0, 1, \ldots, n-1\}$. Suppose rows $\{i_1, i_2, \ldots, i_s\}$ (in order) and columns*

$\{j_1, j_2, \ldots, j_s\}$ are chosen to form an $s \times s$ submatrix S of V and that $\{i_1, i_2, \ldots, i_s\}$ are in arithmetic progression with arithmetic difference k. Then

$$|S| = x_{k_1}^{i_1} x_{k_2}^{i_1} \ldots x_{k_s}^{i_1} |V(x_{k_1}^k, x_{k_2}^k, \ldots, x_{k_s}^k)|$$

Proof: Note that $i_{l+1} - i_l = k$ for $l = 1, 2, \ldots, s-1$, for k the fixed arithmetic difference.

Now $S = \begin{pmatrix} x_{k_1}^{i_1} & x_{k_2}^{i_1} & \ldots & x_{k_s}^{i_1} \\ x_{k_1}^{i_2} & x_{k_2}^{i_2} & \ldots & x_{k_s}^{i_2} \\ \vdots & \vdots & \vdots & \vdots \\ x_{k_1}^{i_s} & x_{k_2}^{i_s} & \ldots & x_{k_s}^{i_s} \end{pmatrix}$ and so $|S| = \begin{vmatrix} x_{k_1}^{i_1} & x_{k_2}^{i_1} & \ldots & x_{k_s}^{i_1} \\ x_{k_1}^{i_2} & x_{k_2}^{i_2} & \ldots & x_{k_s}^{i_2} \\ \vdots & \vdots & \vdots & \vdots \\ x_{k_1}^{i_s} & x_{k_2}^{i_s} & \ldots & x_{k_s}^{i_s} \end{vmatrix}$.

Hence by factoring out x_{k_i} from column i for $i = 1, 2, \ldots, s$ it follows that

$$\begin{aligned} |S| &= x_{k_1}^{i_1} x_{k_2}^{i_1} \ldots x_{k_s}^{i_1} \begin{vmatrix} 1 & 1 & \ldots & 1 \\ x_{k_1}^k & x_{k_2}^k & \ldots & x_{k_s}^k \\ x_{k_1}^{2k} & x_{k_2}^{2k} & \ldots & x_{k_s}^{2k} \\ \vdots & \vdots & \vdots & \vdots \\ x_{k_1}^{(s-1)k} & x_{k_2}^{(s-1)k} & \ldots & x_{k_s}^{(s-1)k} \end{vmatrix} \\ &= x_{k_1}^{i_1} x_{k_2}^{i_2} \ldots x_{k_s}^{i_s} |V(x_{k_1}^k, x_{k_2}^k, \ldots, x_{k_s}^k)| \end{aligned}$$

□

A similar result holds when the columns $\{j_1, j_2, \ldots, j_s\}$ are in arithmetic progression.

Corollary 5.1: *$|S| \neq 0$ if and only if $|V(x_{k_1}^k, x_{k_2}^k, \ldots, x_{k_s}^k)| \neq 0$.*

Corollary 5.2: *$|S| \neq 0$ if and only if $x_{k_i}^k \neq x_{k_j}^k$ for $i \neq j, 1 \leq i, j \leq s$. This happens if and only if $(x_{k_i} x_{k_j}^{-1})^k \neq 1$ for $i \neq j, 1 \leq i, j \leq s$.*

Corollary 5.3: *$|S| \neq 0$ if and only if $(x_{k_i} x_{k_j}^{-1})$ is not a kth root of unity for $i \neq j, 1 \leq i, j \leq s$.*

Corollary 5.4: *When $k = 1$ (that is when consecutive rows are taken) then $|S| \neq 0$.*

Proof: This follows from Corollary 5.3 as $(x_{k_i} x_{k_j}^{-1}) \neq 1$ for $i \neq j$. □

Corollary 5.5: *Let $x_i = \omega^{i-1}$ where ω is a primitive nth root of unity (that is, when V is the Fourier $n \times n$ matrix) and suppose $\gcd(k, n) = 1$. Then $|S| \neq 0$.*

Proof: If $(x_{k_1} x_{k_j}^{-1})^k = 1$ then $(\omega^{k_i - 1} \omega^{1-k_j})^k = 1$ and so $\omega^{k(k_i - k_j)} = 1$. As ω is a primitive nth root of unity this implies that $k(k_i - k_j) \equiv 0 \mod n$. As $\gcd(k, n) = 1$ this implies $k_i - k_j \equiv 0 \mod n$ in which case $k_i = k_j$ as $1 \leq k_i < n, 1 \leq k_j < n$. □

Recall that an mds code is one of the form $(n, r, n - r + 1)$ which attains the maximum distance possible for an (n, r) code. mds codes with efficient decoding algorithm are the goal.

An mds (n, r) code $\mathcal{C}$ is characterised by either of the following equivalent conditions, Blahut (2003):

- $\mathcal{C}$ is an $(n, r, n - r + 1)$ code.
- $\mathcal{C}^{\perp}$ is an mds $(n, n - r, r + 1)$ code, where $\mathcal{C}^{\perp}$ is the dual code of $\mathcal{C}$.
- Any $(n - r)$ columns of a check matrix for $\mathcal{C}$ are linearly independent.
- Any r columns of a generator matrix for $\mathcal{C}$ are linearly independent.

As long as we take the rows of the $n \times n$ Vandermonde matrix in arithmetic sequence k and the entries x_i are such that $(x_i x_j^{-1})$ is not a kth root of unity for $i \neq j$ then mds codes will be generated by these rows. When $k = 1$, in which case consecutive rows of the matrix are taken, then always $\gcd(n, k) = 1$. When the Vandermonde matrix in question is the Fourier matrix in addition it will be shown that practical decoding algorithms exist for these cases.

5.2 *Fourier matrix*

The Fourier matrix is a special type of Vandermonde matrix. Let ω be a primitive nth root of unity in a field $\mathbb{F}$. The Fourier matrix F_n, relative to ω and $\mathbb{F}$, is the $n \times n$ matrix

$$F_n = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)(n-1)} \end{pmatrix}$$

Simplifications can be made to the powers by noting $\omega^n = 1$. Then

$$\begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)(n-1)} \\ 1 & \omega^{n-2} & \omega^{2(n-2)} & \dots & \omega^{(n-1)(n-2)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \omega & \omega^2 & \dots & \omega^{(n-1)} \end{pmatrix} = nI_n$$

The inverse of F_n can be obtained from the above by multiplying through by n^{-1} when it exists. An nth root of unity can only exist in a field provided the characteristic of the field does not divide n and in this case, the n^{-1} exists.

If ω is a primitive nth root of unity then so is ω^k where $\gcd(n, k) = 1$ and in these cases the Fourier matrix may be defined by replacing ω by ω^k to obtain another Fourier matrix. Notice that the second matrix on the left in the above is obtained by replacing ω by ω^{n-1} and is thus also a Fourier matrix (relative to ω^{n-1} and $\gcd(n, n - 1) = 1$).

Denote the rows of F_n in order by $\{E_0, E_1, \dots, E_{n-1}\}$. It is easily checked that $E_i {E_{n-i}}^{\mathrm{T}} = n$ and $E_i {E_j}^{\mathrm{T}} = 0$ for $j \neq n - i \mod n$. Thus

$$\begin{pmatrix} E_0 \\ E_1 \\ \vdots \\ E_{n-1} \end{pmatrix} ({E_0}^{\mathrm{T}}, {E_{n-1}}^{\mathrm{T}}, {E_{n-2}}^{\mathrm{T}}, \dots, {E_1}^{\mathrm{T}}) = nI_n$$

Call this the *Fourier Equation* for future reference. We are assuming the Fourier matrix exists over the field and in particular, any r rows or any r columns are linearly independent.

Suppose then the first r rows of F_n are used to form a generating matrix A for a (n, r) code $\mathcal{C}_r$. Now using the unit-derived scheme from the Fourier matrix we see that

$$\begin{pmatrix} E_0 \\ E_1 \\ \vdots \\ E_{r-1} \end{pmatrix} ({E_{n-r}}^{\mathrm{T}}, {E_{n-2}}^{\mathrm{T}}, \ldots, {E_1}^{\mathrm{T}}) = 0_{n-r}$$

which corresponds to $AD = 0_{n-r}$ where D^{T} is a check matrix. Thus a check matrix is

$$\begin{pmatrix} E_{n-r} \\ E_{n-r-1} \\ \vdots \\ E_1 \end{pmatrix} \text{ and hence } \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_{n-r} \end{pmatrix} \text{ is a check matrix.}$$

Suppose a codeword αA is transmitted but $\alpha A + w$ with error w is received where w is an $1 \times n$ vector. Then $< E_i, w >= \alpha_i$ are known for $i = 1, 2, \ldots (n - r)$ since $(\alpha A + w){E_i}^{\mathrm{T}} = w{E_i}^{\mathrm{T}} =< w, E_i >$ for these i.

The star multiplication, $*$, is explained further in Section 6.1 but is simply multiplying corresponding entries of vectors: If x_i denotes the ith component of a vector $\underline{x}$ in $\mathbb{F}^n$ then $\underline{a} * \underline{b}$ for $\underline{a}, \underline{b} \in \mathbb{F}^n$ is defined to be the vector with components $a_i * b_i$ in ith position. The rows of F_n also have the nice property that $E_i * E_j = E_{i+j}$ where suffices are taken $\bmod n$ and this is very useful for describing error-correcting algorithms.

5.3 Consecutive rows

First of all, consider cases where consecutive rows of the Vandermonde matrix are taken to define a unit-derived code.

The Vandermonde matrix is

$$V = V(x_1, x_2, \ldots, x_n) = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_n \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \ldots & x_n^{n-1} \end{pmatrix}$$

This has inverse U with $VU = I_n$. When V is a Fourier matrix the inverse matrix U of V is easy to find and can be written down directly.

Let A be the matrix of the first r rows of V and D the matrix of the last $(n - r)$ columns of U. By unit-derived scheme then, $AD = 0$ and D^{T} is the check matrix of the (n, r) code $\mathcal{C}$ generated by A. Now $\mathcal{C}^{\perp}$ is the dual code of $\mathcal{C}$ and is generated by the rows of D^{T}. It is known that $\mathcal{C}$ is an mds code if and only if $\mathcal{C}^{\perp}$ is an mds code.

Proposition 5.2: *Any $r \times r$ submatrix of A is a Vandermonde matrix $V(x_{i_1}, x_{i_2}, \ldots, x_{i_r})$ for $i_j \in \{1, 2, \ldots, n\}$ with $i_1 < i_2 < \ldots < i_r$.*

Proof: This follows from Proposition 5.1 above.

□

Corollary 5.6: *Any $r \times r$ submatrix of A has* $\det \neq 0$.

Corollary 5.7: *The code $\mathcal{C}^{\perp}$ is an mds code.*

Proof: This is true since A is the check matrix of $\mathcal{C}^{\perp}$ and every $r \times r$ submatrix of A has non-zero determinant so that the minimum distance of the $(n, n-r)$ code $\mathcal{C}^{\perp}$ is $r+1$. □

Corollary 5.8: *The code $\mathcal{C}$ is an $(n, r, n-r+1)$ mds code.*

Proof: This is because $\mathcal{C}^{\perp}$ is an mds $(n, n-r, r+1)$ code. It may also be seen from the fact that any r columns of A are linearly independent since the determinant of any $r \times r$ submatrix of A is $\neq 0$. □

Take any r consecutive rows of a Vandermonde matrix as follows:

$$A = \begin{pmatrix} x_1^{r_1} & x_2^{r_1} & \dots & x_n^{r_1} \\ x_1^{r_1+1} & x_2^{r_1+1} & \dots & x_n^{r_1+1} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{r_1+r-1} & x_2^{r_1+r-1} & \dots & x_n^{r_1+r-1} \end{pmatrix}$$

Write i_j for x_{i_j}. Now any $r \times r$ submatrix of A has the form

$$\begin{pmatrix} i_1^{r_1} & i_2^{r_1} & \dots & i_r^{r_1} \\ \vdots & \vdots & \vdots & \vdots \\ i_1^{r_1+r-1} & i_2^{r_1+r-1} & \dots & i_r^{r_1+r-1} \end{pmatrix}.$$

The determinant of this is by Proposition 5.1

$$i_1^{r_1} i_2^{r_1} \dots i_r^{r_1} \begin{vmatrix} 1 & 1 & \dots & 1 \\ i_1 & i_2 & \dots & i_r \\ \vdots & \vdots & \vdots & \vdots \\ i_1^{r-1} & i_2^{r-1} & \dots & i_r^{r-1} \end{vmatrix} = i_1^{r_1} i_2^{r_1} \dots i_r^{r_1} |V(i_1, i_2, \dots, i_r)|.$$

This is clearly non-zero - we are assuming the x_j are distinct and non-zero.

This gives further mds codes from the unit scheme.

Proposition 5.3: *Let $\mathcal{C}_r$ be a code obtained by taking any r rows in succession of a Vandermonde $n \times n$ matrix as a generator matrix. Then $\mathcal{C}_r$ is an mds $(n, r, n-r+1)$ code.*

5.4 Rows in an arithmetic sequence

Now choose r rows in sequence with the same arithmetic difference p. Consider the case where the sequence starts at the first row; cases where the sequence begins at another row are similar. Then the matrix formed is

$$A = \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1^p & x_2^p & \dots & x_n^p \\ x_1^{2p} & x_2^{2p} & \dots & x_n^{2p} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{p(r-1)} & x_2^{p(r-1)} & \dots & x_n^{p(r-1)} \end{pmatrix}.$$

Here we begin at the first row and assume $p(r-1) \leq n$. It may be possible to overlap and take $p*j$ to be $p*j \mod n$, and the added assumption that $r < n$. In particular, overlapping is possible when the Vandermonde unit schemes consist of Fourier matrices.

The check matrix is obtained by deleting the corresponding columns of the inverse of V.

Any $r \times r$ submatrix of A has the form

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ i_1^p & i_2^p & \dots & i_r^p \\ i_1^{2p} & i_2^{2p} & \dots & i_r^{2p} \\ \vdots & \vdots & \vdots & \vdots \\ i_1^{p(r-1)} & i_2^{p(r-1)} & \dots & i_r^{p(r-1)} \end{pmatrix}$$

where i_j^k means $x_{i_j}^k$. This has determinant $\prod_{k<j}(i_k^p - i_j^p)$. It is easy to decide when this is non-zero.

This determinant is non-zero if and only for all $i_k, i_j, k \neq j$ that $i_k^p - i_j^p \neq 0$ and this happens if and only if $(i_k i_j^{-1})^p \neq 1$ which happens if and only if $i_k i_j^{-1}$ is not a pth root of unity.

From Corollary 5.5 it is noted that when $\gcd(n,k) = 1$ and the Vandermonde matrix is a Fourier matrix then the determinant is never 0. This gives the following proposition.

Proposition 5.4: *Let F be a Fourier $n \times n$ matrix. Suppose a code is obtained from F by choosing in order r rows which are in arithmetic sequence k with $\gcd(n,k) = 1$ to form the generator matrix of a code. Then the code is an mds $(n, r, n-r+1)$ code.*

Note also for the Fourier matrix that it is possible to *overlap* in selection and still obtain an mds code.

Proposition 5.5: *Let $V = V(x_1, x_2, \dots, x_n)$ be a Vandermonde $n \times n$ matrix such that $x_i x_j^{-1}$ is not a kth root of unity for any $i \neq j$. Suppose a code is obtained by choosing in order r rows from V which are in arithmetic sequence k to form a code. Then the code is an mds $(n, r, n-r+1)$ code.*

6 Decoding

The following decoding methods are sourced from Hurley (2017) which is an application of Pellikaan's decoding method using error correcting pairs Pellikaan (1992) when such exist.

Error correcting pairs were introduced by Pellikaan Pellikaan (1992) and Duursma & Kötter Duursma and Kötte (1994). The method of Pellikaan is found more useful here and in Hurley (2017); the decoding algorithm of Pellikaan has a precise translation into a linear algebra method for the codes constructed here as explained in Section 3 of Hurley (2017).

6.1 Preliminaries

First, some preliminaries are required. Let F be a field and $\mathcal{C}$ a (linear) code over F. Write $n(\mathcal{C})$ for the code length of $\mathcal{C}$, its minimum distance is denoted by $d(\mathcal{C})$ and denote its dimension by $k(\mathcal{C})$.

Now w_i denotes the ith component of $w \in F^n$. For any $w \in F^n$ define the support of w by $\text{supp}(w) = \{i | w_i \neq 0\}$ and the zero set of w by $z(w) = \{i | w_i = 0\}$. The weight of w is the number of non-zero coordinates of w and denote it by $wt(w)$. The number of elements of a set I is denoted by $|I|$. Thus $wt(a) = |\operatorname{supp}(w)|$.

We say that w has t errors supported at I if $w = c + e$ with $c \in \mathcal{C}$ and $I = \text{supp}(e)$ and $|I| = t = d(w, \mathcal{C})$.

The bilinear form $<,>$ is defined by $< a, b >= \sum_i a_i b_i$. For a subset C of F^n, the dual $C^\perp$ of C in F^n with respect to the bilinear form $<,>$ is defined by $C^\perp = \{x | < x, c >= 0, \forall c \in C\}$.

As usual, the sum of two elements of F^n is defined by adding corresponding coordinates. Of use in these considerations is what is termed the *star multiplication* $a * b$ of two elements $a, b \in F^n$ defined by multiplying corresponding coordinates, that is $(a * b)_i = a_i b_i$. For subsets A and B of F^n denote the set $\{a * b | a \in A, b \in B\}$ by $A * B$. If A is generated by X and B is generated by Y then $A * B$ is generated by $X * Y$.

Definition 6.1: Let A, B and C be linear codes in F^n. We call (A, B) a *t-error correcting pair for C* if
1) $A * B \subseteq C^\perp$
2) $k(A) > t$
3) $d(A) + d(C) > n$,
4) $d(B^\perp) > t$.

For more information on this consult Pellikaan (1992).

Consider now a Fourier $n \times n$ matrix. It is shown below that error-correcting pairs exist for codes generated by the rows of this Fourier matrix where the rows are taken in succession or in arithmetic sequence k with $\gcd(n, k) = 1$.

Let $F = F_n$ be a Fourier $n \times n$ matrix with ω as the element of order n.

Denote the rows of F in order by $\{E_0, E_1, \ldots, E_{n-1}\}$. It is easily checked that $E_i E_{n-i}{}^{\mathrm{T}} = n$ and $E_i E_j{}^{\mathrm{T}} = 0$ for $j \neq n - i \mod n$. Thus

$$\begin{pmatrix} E_0 \\ E_1 \\ \vdots \\ E_{n-1} \end{pmatrix} (E_0{}^{\mathrm{T}}, E_{n-1}{}^{\mathrm{T}}, E_{n-2}{}^{\mathrm{T}}, \ldots, E_1{}^{\mathrm{T}}) = nI_n.$$

Call this the *Fourier Equation* for future reference.

Note that if H is a check matrix for a code then also αH is a check matrix for the code for any $\alpha \neq 0$.

We write out the details for the cases where the first r rows are taken as the generator matrix. The cases where rows are taken in succession or where rows are taken in arithmetic sequence k with $\gcd(n,k)=1$ are similar; in all cases, it requires getting error-correcting pairs and working from there.

The general Vandermonde case with a restriction on cases where the rows are taken in an arithmetic sequence, is given in Section 5.1.

Suppose then $\mathcal{C}$ is the code obtained by taking the first r rows of F. Thus $\mathcal{C} = \langle E_0, E_1, \ldots, E_{r-1}\rangle$. Then $\mathcal{C}^\perp$ is $\langle E_1, E_2, \ldots, E_{n-r}\rangle$ which can also be obtained by eliminating the first r columns of the second matrix on the left in the Fourier Equation.

Note that $E_i * E_j = E_{i+j}$ where suffices are taken $\mod n$. Let $A = \langle E_1, E_2 \ldots, E_{t+1}, \rangle, B = \langle E_0, E_1, \ldots, E_{t-1}\rangle$ when $(n-r)$ is even and $t = \frac{n-r}{2}$, and let $A = \langle E_1, E_2, \ldots, E_{t+1}\rangle, B = \langle E_0, E_1, \ldots, E_t\rangle$ when $(n-r)$ is odd and $t = \lfloor \frac{n-r}{2} \rfloor$.

Then it may be verified that A, B is a t-error correcting pair for $\mathcal{C}$.

Thus:

- $A * B \subseteq \mathcal{C}^\perp$
- $k(A) > t$
- $d(A) + d(\mathcal{C}) > n$
- $d(B^\perp) > t$

This gives the following algorithm for locating and quantifying up to t errors for the code $\mathcal{C}$. In Hurley (2017) the method of error-correcting pairs of Pellikaan Pellikaan (1992) is translated into an algorithm for decoding codes defined by rows in succession or in (certain) arithmetic sequences of a Vandermonde/Fourier matrix. This may be applied directly here.

Let C be the $r \times n$ generator matrix of $\mathcal{C}$. Suppose now α is a $1 \times r$ codeword, that αC is sent but that $\alpha C + \epsilon$ is received for a $1 \times n$ vector ϵ with at most t non-zero entries.

Assume $(n-r)$ is even; the other case is similar. Thus we are assuming $n - r = 2t$. Now $\mathcal{C}^\perp$ is a check matrix for the code and thus $\epsilon E_1, \epsilon E_2, \ldots, \epsilon E_{n-r}$ are known by applying the check matrix to $\alpha C + \epsilon$. Let $\alpha_i = \epsilon E_i$ for $i = 1, 2, \ldots, n-r (= 2t)$.

The algorithm then is:

Algorithm 6.1:

1. *Find a non-zero solution of the kernel of the* $t \times (t+1)$ *Hankel matrix*
$$\begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \ldots, & \alpha_{t+1} \\ \alpha_2 & \alpha_3 & \alpha_4 & \ldots & \alpha_{t+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_t & \alpha_{t+1} & \alpha_{t+2} & \ldots & \alpha_{2t} \end{pmatrix}.$$
Call this solution $\underline{x}^T$ *which is a* $(t+1) \times 1$ *vector.*

2. *Let* $\underline{a} = (E_1, E_2, \ldots, E_{t+1})\underline{x}^T$ *which is a* $1 \times n$ *vector.*
(Any non-zero multiple of $\underline{a}$ *will suffice as we are only interested in the zero entries of* $\underline{a}$*. Note that* $\underline{a}$ *is a* $1 \times n$ *vector.)*

3. *Let* $z(\underline{a}) = \{j | a_j = 0\}$ *which is the set of locations of the zero coordinates of* $\underline{a}$*.*
Suppose $z(\underline{a}) = \{j_1, j_2, \ldots, j_t\}$ *and denote this set by* J*.*

4. *Solve $s_J(x) = s(w)$. This reduces to solving the following. Here $E_i = (E_{i,1}, E_{i,2}, \dots, E_{i,n})$.*

$$\begin{pmatrix} E_{1,j_1} & E_{1,j_2} & \dots & E_{1,j_t} \\ E_{2,j_1} & E_{2,j_2} & \dots & E_{2,j_t} \\ \vdots & \vdots & \vdots & \vdots \\ E_{2t,j_1} & E_{2t,j_2} & \dots & E_{2t,j_t} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2t} \end{pmatrix} \tag{1}$$

5. *Now since in this case $E_{i,j} = \omega^{i*j}$ the equation 1 may be put in the form*

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ \omega^{j_1} & \omega^{j_2} & \dots & \omega^{j_t} \\ \omega^{2j_1} & \omega^{2j_2} & \dots & \omega^{2j_t} \\ \vdots & \vdots & \vdots & \vdots \\ \omega^{(2t-1)j_1} & \omega^{(2t-1)j_2} & \dots & \omega^{(2t-1)j_t} \end{pmatrix} \begin{pmatrix} \omega^{j_1}x_1 \\ \omega^{j_2}x_2 \\ \vdots \\ \omega^{j_t}x_t \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2t} \end{pmatrix} \tag{2}$$

(This form shows that the equation to be solved is a Vandermonde system containing roots of unity but not a (full) Fourier matrix.)

6. *The value of w is then the solution of equations (1) or equivalently equations (2) with entries in appropriate places as determined by J.*

6.2 *In the arithmetic sequence*

Suppose A is an $n \times n$ Fourier matrix with rows $\{E_0, E_1, \dots, E_{n-1}\}$; these rows satisfy $E_i * E_j = E_{i+j}$.

The $E_j w$ are known for $j \in J = \{j_1, j_2, \dots, j_u\}$ where $u \geq 2t$. The elements in J are in arithmetic progression with difference k satisfying $\gcd(n, k) = 1$. Then w is calculated by the following algorithm. Let $\alpha_k =< w, F_{j_k} >= F_{j_k} w$ for $j_k \in J$. Define $F_i = E_{j_i}$ for $j_i \in J$ and $F_0 = E_{j_1 - k}$ with indices taken $\mod n$. Let $F_i = (F_{i,1}, F_{i,2}, \dots, F_{i,n})$.

Algorithm 6.2:

i. *Find a non-zero element x^T of the kernel of* $E = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{t+1} \\ \alpha_2 & \alpha_3 & \dots & \alpha_{t+2} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_t & \alpha_{t+1} & \dots & \alpha_{2t} \end{pmatrix}$.

ii. *Let $\underline{a} = (F_0, F_1, \dots, F_t)x^T$. (Any non-zero multiple of a will suffice as we are only interested in the zero entries of $\underline{a}$. Note that $\underline{a}$ is a $1 \times n$ vector.)*

iii. *Let $z(\underline{a}) = \{j | a_j = 0\}$ which is the set of locations of the zero coordinates of $\underline{a}$. Suppose $z(\underline{a}) = \{j_1, i_2, \dots, j_t\}$ and denote this set by J.*

iv. *Solve $s_J(x) = s(w)$. This reduces to solving the following:*

$$\begin{pmatrix} F_{1,j_1} & F_{1,j_2} & \dots & F_{1,j_t} \\ F_{2,j_1} & F_{2,j_2} & \dots & F_{2,j_t} \\ \vdots & \vdots & \vdots & \vdots \\ F_{2t,j_1} & F_{2t,j_2} & \dots & F_{2t,j_t} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2t} \end{pmatrix} \tag{3}$$

v. *The value of w is then the solution of these equations with entries in appropriate places as determined by J.*

In Algorithm 6.1 it is shown that the equation (1) are equivalent to a Vandermonde system of equation (2); similarly, here it can be seen that the equation in (3) are equivalent to a Vandermonde system with roots of unity as entries (but not the full Fourier matrix).

6.3 The general Vandermonde case

Working with a general Vandermonde matrix introduces difficulties as the inverse is not always nice to work with. However error-correcting algorithms can be formulated in many cases and we briefly discuss these cases here.[5]

Consider the Vandermonde matrix

$$V = V(\beta_1, \beta_2, \dots, \beta_n) = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \beta_1 & \beta_2 & \dots & \beta_n \\ \vdots & \vdots & \vdots & \vdots \\ \beta_1^{n-1} & \beta_2^{n-1} & \dots & \beta_n^{n-1} \end{pmatrix}$$

We assume the β_i are distinct and non-zero.

Denote the rows of V in order by $\{E_0, E_1, \dots, E_{n-1}\}$. Then $E_i * E_j = E_{i+j}$ as long as $i + j \leq n$.

Define E_k to be $(\beta_1^k, \beta_2^k, \dots, \beta_n^k)$ for any $k \in \mathbb{Z}$. The rows of V are $\{E_0, E_1, \dots, E_{n-1}\}$ and these have been extended.

Lemma 6.1: $E_i * E_j = E_{i+j}$.

Proof: This is simply because $\beta^i \beta^j = \beta^{i+j}$. □

Let $\mathcal{C}^\perp = \langle E_{j_1}, E_{j_2}, \dots, E_{j_u} \rangle$, where $u = 2t$. If $\mathcal{C}^\perp$ has rows in arithmetic sequence with arithmetic difference k and the ratios $\beta_i \beta_i^{-1}$ for $i \neq j$ in V are not kth roots of unity then $\mathcal{C}$ (the dual of $\mathcal{C}^\perp$) is an $(n, n-2t, 2t+1)$ code, see Proposition 5.5, and is t-error correcting with $C^\perp$ as the check matrix. Then also $\mathcal{C}$ has an error correcting pair and a decoding Algorithm may be derived. However, it is not easy to describe $\mathcal{C}$ itself for this general Vandermonde case.

Let $\alpha_i =< w, E_{j_i} >= E_{j_i} w^{\mathrm{T}}$ for $j_i \in J$. Let $F_i = E_{j_i}$ for $j_i \in J$. Thus $\alpha_i =< w, F_i >$.

Algorithm 6.3:

(i) Find a non-zero element v^T of the kernel of $E = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots \alpha_{t+1} \\ \alpha_2 & \alpha_3 & \dots \alpha_{t+2} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_t & \alpha_{t+1} & \dots & \alpha_{2t} \end{pmatrix}$.

(ii) Let $a = (F_1, F_2, \dots, F_{t+1})v^T$.

(iii) Let $z(a) = \{j | a_j = 0\}$ which is the set of locations of the zero coordinates of a. Suppose $z(a) = \{i_1, i_2, \dots, i_t\}$ and denote this set by J.

(iv) Solve $s_J(x) = s(w)$. This reduces to solving the following:

$$\begin{pmatrix} \beta_{i_1}^{j_1} & \beta_{i_2}^{j_1} & \dots & \beta_{i_t}^{j_1} \\ \beta_{i_1}^{j_2} & \beta_{i_2}^{j_2} & \dots & \beta_{i_t}^{j_2} \\ \vdots & \vdots & \vdots & \vdots \\ \beta_{i_1}^{j_{2t}} & \beta_{i_2}^{j_{2t}} & \dots & \beta_{i_t}^{j_{2t}} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2t} \end{pmatrix} \tag{4}$$

Since the entries in the matrix of (4) have arithmetic difference k giving that $j_s = i_1 + (s-1)k$ for $1 \le s \le 2t$, the equation (4) is equivalent to

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ \beta_{i_1}^{k} & \beta_{i_2}^{k} & \dots & \beta_{i_t}^{k} \\ \vdots & \vdots & \vdots & \vdots \\ \beta_{i_1}^{(2t-1)k} & \beta_{i_2}^{(2t-1)k} & \dots & \beta_{i_t}^{(2t-1)k} \end{pmatrix} \begin{pmatrix} \beta_{i_1}^{j_1} x_1 \\ \beta_{i_2}^{j_1} x_2 \\ \vdots \\ \beta_{i_t}^{j_1} x_t \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2t} \end{pmatrix} \tag{5}$$

(v) Then $x = (x_1, x_2, \dots, x_t)$ is obtained from these equations (5) (or from (4)) and w has entries x_i in positions as determined by J and zeros elsewhere.

The matrix in equation (5) is a Vandermonde matrix. It is sufficient to solve the first t equations and the inverse of such a $t \times t$ Vandermonde type matrix may be obtained in $O(t^2)$ operations. In connection with item 6.3, finding a non-zero element of the kernel of a Hankel $t \times (t+1)$ matrix can be done in $O(t^2)$ or less operations.

7 Code to a rate and error capability

Suppose an mds code of rate $R = \frac{r}{n}$ is required.

It is required to obtain over a finite field a Fourier $n \times n$ matrix.

We can take n to be as large as necessary as $\frac{r}{n} = \frac{rs}{sn}$ for any positive integer s.

Let p be a prime not dividing n. Then by Euler's theorem, $p^{\phi(n)} \equiv 1 \mod n$ where ϕ is the Euler ϕ function. Thus $p^{\phi(n)} - 1 = nq$ for some positive integer q. Consider the field $\mathbb{F} = GF(p^{\phi(n)})$. Then a primitive generator, β say, of the field has order $(p^{\phi(n)} - 1) = nq$. Then $\beta^q = \omega$ has order n in $\mathbb{F}$. Construct the Fourier $n \times n$ matrix, F_n, over $\mathbb{F}$ using ω as the element of order n. Now by the method of the previous sections, $(n, r, n-r+1)$ codes can be constructed with efficient decoding algorithms from F_n.

For a prime p not dividing n, we know that there exists a positive integer q such that $p^q \equiv 1 \mod n$. So for best results take q to be the smallest such positive integer and do the calculations in $GF(p^q)$.

If n is odd then $2 \not| n$ and so the Fourier matrix can be obtained over $GF(2^k)$ for some k, where $2^k \equiv 1 \mod n$. For example, if $n = 103$ then the order of 2 mod 103 is 51 and so the Fourier matrix may be obtained over $GF(2^{51})$. Making the calculations over $GF(2^s)$ has advantages in that codes over such a field may be transmitted as binary digits.

Suppose a rate $R = \frac{r}{n}$ is required and in addition, t errors may need to be corrected. Then it is required that $t = \lfloor \frac{n-r}{2} \rfloor$. Assume $n - r$ is even; the other case is similar. Then it is required that $t = \frac{n-r}{2} = \frac{n(1-R)}{2}$.

7.1 Examples

7.1.1 Rate $\frac{5}{7}$

Suppose a rate of $\frac{5}{7}$ is required and that $t = 50$ errors should be correctable. This gives that $\frac{n(1-\frac{5}{7})}{2} = t = 50$ which requires $n = 350$. Thus a code $(350, 250, 101)$ is required. Thus construct a Fourier 350×350 matrix over a field. Now 3 is a prime not dividing $n = 350$ and the order of $3 \mod 350$ is 60. Thus this required Fourier matrix exists over $GF(3^{60})$. Also, the order of $11 \mod 350$ is 15 and the field $GF(11^{15})$ may also be used. A little investigation shows that the order of $43 \mod 350$ is 4 so the field $GF(43^4)$ could be used.

Let the required rate again be $\frac{5}{7}$ and now it is required that $t = 49$ errors be correctable. This gives that $\frac{n(1-\frac{5}{7})}{2} = t = 49$ which requires $n = 343$. Require a $(343, 245, 99)$ code. Since n is odd it is possible to find a field $GF(2^s)$ which has a 343rd root of unity. The order of $2 \mod 343$ is 147 so the field $GF(2^{147})$ could be used but would be large. However, the order of $19 \mod 343$ is 6 so it is possible to work in $GF(19^6)$.

Let the required rate again be $\frac{5}{7}$ and now it is required that $t = 48$ errors be correctable. This gives that $\frac{n(1-\frac{5}{7})}{2} = t = 48$ which requires $n = 336$. Require a $(336, 240, 97)$ code. Now note that 337 is prime so can work in the prime field $GF(337)$ which involves modular arithmetic. An element of order 336 is required in $GF(337)$ and this is easily found. For example, the order of $10 \mod 337$ is 336 and thus $\omega = 10 \mod 337$ may be used as the element of order 336 in forming the Fourier 336×336 matrix over $GF(337)$. Here the arithmetic is modular arithmetic, which is nice.

7.1.2 Rate $\frac{31}{32}$

Suppose a rate $\frac{31}{32}$ is specified and we would like the code to correct at least 50 errors. Then for $(n, r, n - r + 1)$ we need $n - r \geq 2 * 50 = 100$ and so need for $R = \frac{31}{32}$ that $n * \frac{1}{32} \geq 100$ which is $n \geq 3200$. We would also like to work with modular arithmetic. Now notice that 3201 is not a prime but that 3203 is a prime. Thus let $n = 3202$ and construct the code $(3202, 3102, 101)$ over the prime field $GF(3203)$. This code has rate slightly less (0.0000019..) than $\frac{31}{32}$. To have rate of $\frac{31}{32}$ and still work over a prime field take $n = 104 * 32 = 3328$ and then $n + 1 = 3329$ is prime. Here we work over the prime field $GF(3329)$ and get the code $(3328, 3224, 105)$ which can correct 52 errors.

The order of $2 \mod 3203$ is 3220 so $2 \mod 3203$ may be used in as the element of order 3202 for the Fourier 3202×3202 matrix in $GF(3203)$. All the non-zero elements of $GF(3203)$ are used for this Fourier matrix. From it codes of all forms $(3203, r, 3203 - r + 1)$ may be obtained for $1 \leq r \leq 3202$.

7.2 Rate $\frac{3}{4}$; correct lots

Suppose, for example, a code is required that could correct 50 errors and have a rate of $\frac{3}{4}$. The code of smallest length satisfying these conditions is one of the form $(400, 300, 101)$, How could such a code be constructed? One way is to construct a Fourier 400×400 matrix and select three quarters of the rows in order so that an mds code is generated. Thus select 300 rows in sequence from the Fourier matrix. What is the smallest field over which such a 400×400 Fourier can exist? What is the field of a smallest characteristic over which such a Fourier matrix can exist? Now $\phi(400) = 160$ so we need the smallest field $GF(p^s)$ such that $p^s \equiv 1 \mod 400$ with $\gcd(400, p) = 1$ and $s|160$ as necessary requirements. Here it is found that $7^4 \equiv 1 \mod 400$ so we can use the field $GF(7^4)$. This is the smallest field for which there exists a 400×400 Fourier matrix. Now $7^4 = 2401$ and thus the field is relatively small and its characteristic is small. The 400×400 Fourier matrix over $GF(7^4)$ can be used to find the $(400, 300, 101)$ code but it can also be used to find $(400, r, 401 - r)$ codes over $GF(7^4)$. For example $(400, 350, 51)$ code can correct 25 errors and $(400, 200, 201)$ code over $GF(7^4)$ can correct 100 errors.

Consider constructing a code of rate $\geq \frac{3}{4}$ and which can correct 50 errors but now require the code to be over $GF(p)$ for a prime p. Now the order of the non-zero elements of $GF(p)$ is $p - 1$ and we require $n = p - 1 \geq 400$. It turns out that $p = 401$ is a prime which is the least prime p for which $p \geq 400$. Now the order of $2 \mod 401$ is 200 so using $2 \in GF(401)$ doesn't work but the order of $3 \mod 401$ is 400. Hence let $\omega = 3 \mod 401$ and form the 400×400 Fourier matrix F_{400} over $GF(401)$ with ω as a primitive 400th root of unity. Now choose the first 300 rows of F_{400} or any consecutive 300 rows in F_{400} gives a $(400, 300, 101)$ code as required.

The calculations are done $\mod 401$. Error correcting pairs are also obtainable from the unit Fourier scheme which are then used for the efficient decoding algorithms.

Which are better, the codes over $GF(7^4)$ or the codes over $GF(401)$?

7.3 Remark

Many such constructions are possible. Codes over $GF(2^s)$, for s not too large, and codes over prime fields may be particularly useful.

7.4 'Optimal' codes from a given field

Suppose the field $GF(p^s)$ is given and it is required to construct the best possible codes with coefficients from this field. Let $n = p^s - 1$. Then there exists an element ω of order n in $GF(p^s)$ and every non-zero element is a power of this generator. Form the Fourier $n \times n$ matrix using ω as a primitive nth root of unity. Unit-derived codes are then formed using rows of F in succession or else in arithmetic sequence k satisfying $\gcd(n, k) = 1$. For any $1 \leq r \leq n$, mds codes of the form $(n, r, n - r + 1)$ may be constructed from the rows of this Fourier matrix. The Fourier matrix uses all the non-zero elements of $GF(p^s)$.

These are the best performing codes from $GF(p^s)$; the lengths are $p^s - 1$ and all possible rates $\frac{r}{n}$ with $r \leq n$ are available.

8 Shannon

Here we relate the previous Hamming results to Shannon results.

For a given rate $1 \geq R > 0$ the previous sections give methods for constructing $(n, r, n-r+1)$ codes where $\frac{r}{n} = R$. The probability of error is the probability that more than $k = \lfloor \frac{n-r}{2} \rfloor$ errors occur in the binomial distribution with p the probability that an error occurs at a component. Here $\mu = np$.

Chernoff's bounds Chernoff (1952) give the following:

$$\mathsf{P}[X \geq (1+\delta)\mu] \leq (\frac{e^{\delta}}{(1+\delta)^{1+\delta}})^{\mu} \leq e^{\frac{-\delta^2}{2+\delta}\mu} = e^{\frac{-\delta^2}{2+\delta}np} \text{ for } \delta > 0.$$
$$\mathsf{P}[X \leq (1-\delta)\mu] \leq (\frac{e^{-\delta}}{(1-\delta)^{1-\delta}})^{\mu} < (\frac{e^{-\delta}}{e^{-\delta+\delta^2/2}})^{\mu} < e^{-\delta^2\mu/2} \text{ for } 0 < \delta \leq 1.$$

Now consider a code $(n, r, n-r+1)$ which can correct $k = \lfloor \frac{n-r}{2} \rfloor$ errors and has an efficient decoding algorithm. Assume $n-r$ is even; the other case is similar; thus $k = \frac{n-r}{2}$. Now $r = nR$ where R is the rate.

For the first Chernoof inequality to hold it is required that $(1+\delta)np = k+1 = 1 + \frac{n-r}{2} = 1 + \frac{n(1-R)}{2}$. Thus $(1+\delta) = \frac{1}{np} + \frac{1-R}{2p}$ and thus $\delta = \frac{1-R}{2p} - 1 + \frac{1}{np}$. We require $\delta > 0$ and so require $\frac{1-R}{2p} - 1 + \frac{1}{np} > 0$. Multiply across by $2p$ and this requires $(1 - R) + 2/n > 2p$ which is equivalent to $R < 1 - 2p + 2/n$. For n large enough make $R < 1 - 2p + 2/n$. Then the probability of error is $< e^{\frac{-\delta^2}{2+\delta}np}$.

Now $R < 1 - 2p + \frac{2}{n}$ means that R can be as close to $1 - 2p$ as necessary and then the probability of error is less than $e^{-\gamma n}$ for some $\gamma > 0$. Note that $p < \frac{1}{2}$ implies that $1 - 2p > 0$ and then $R > 0$ also for n big enough.

For the second Chernoff inequality to hold requires $(1-\delta)\mu = \frac{n-r}{2}$ which is $(1 - \delta)np = \frac{n(1-R)}{2}$; this requires $(1-\delta)\frac{1-R}{2p}$ and hence $-\delta = \frac{1-R}{2p} - 1$. Now $\delta > 0$ requires $\frac{1-R}{2p} - 1 < 0$ in which case require $R > 1 - 2p$. For $\delta \leq 1$ requires $-\delta \geq -1$ in which case it is required that $\frac{1-R}{2p} - 1 \geq -1$ from which it is required that $\frac{1-r}{2p} \geq 0$ from which it is required that $1 \geq R$, which is true. Thus the second Chernoff inequality can be applied for $R > 1 - 2p$. Thus for $R > 1 - 2p$ the probability of no error is less than $e^{-\delta^2\mu/2}$. Thus for n big enough the probability of error is $\geq \frac{1}{2}$.

In order to construct a $(n, r, n-r+1)$ code over a finite field by the unit-derived method with Fourier/Vandermonde matrices it is necessary to have a field $F = GF(p^k)$ such that $n|(p^k - 1)$. The rate is $R = \frac{r}{n}$ and n can be taken to be as large as necessary as $\frac{r}{n} = \frac{rs}{sn}$ for any positive integer s.

Let p be a prime not dividing n. Then by Euler's theorem, $p^{\phi(n)} \equiv 1 \mod n$ where ϕ is the Euler ϕ function. Thus $p^{\phi(n)} - 1 = nq$ for some positive integer q. Consider the field $\mathbb{F} = GF(p^{\phi(n)})$. Then a primitive generator β of the field has order $p^{\phi(n)} - 1$. Then $\beta^q = \omega$ has order n in F. Construct the Fourier $n \times n$ matrix, F_n, over $\mathbb{F}$ using ω as the element of order n. Now by the method of the previous sections, $(n, r, n-r+1)$ codes can be constructed with efficient decoding algorithms from F_n.

For a prime p not dividing n we know that there exists a positive integer q such that $p^q \equiv 1 \mod n$. So for best results take q to be the smallest such positive integer and do the calculations in $GF(p^q)$.

If a 'rate' H is required which is not a rational number then take the 'nearest' rational number to H.

9 Complexity

The decoding calculations require finding a non-zero element in the kernel of a Hankel $t \times (t+1)$ matrix. Finding the kernel of an $t \times (t+1)$ Hankel matrix can be done in $O(t^2)$ operations. Super-fast algorithms of $O(t \log^2 t)$ have been proposed with which to find the kernel of a Hankel $t \times (t+1)$ matrix.

It is then required to solve a system of $2t \times t$ equations where the coefficients on the left of the matrix are roots of unity; solving the first $t \times t$ equations is sufficient. The matrix of the system of $t \times t$ equations reduces to a Vandermonde matrix whose entries are roots of unity. Now the system can be solved in $O(t^2)$ operations. The entries of the Vandermonde matrix are roots of unity in a finite field which make the calculations easier and stable.

Consider the case where the encoder is the first part (first rows) of a Fourier matrix. Thus we are in the situation $\begin{pmatrix} A \\ B \end{pmatrix} (C, D) = I$ where $\begin{pmatrix} A \\ B \end{pmatrix}$ is a Fourier matrix and (C, D) is a multiple ($\frac{1}{n}$ for length n) of a Fourier matrix.

The encoding is $\alpha \mapsto \alpha A$ where A is part of a Fourier matrix $F = \begin{pmatrix} A \\ B \end{pmatrix}$. Thus by adding 0^s to the end of α to get $\bar{\alpha}$ of length n ensures the encoding can be done by (Fast) Fourier Transform if necessary.

Similarly the decoding can be done by (Fast) Fourier Transform when the errors have been eliminated as $\alpha AC = \alpha$ and C is part of a Fourier matrix (C, D). In fact $\alpha A(C, D) = (\alpha AC, \alpha AD) = (\alpha, 0)$.

Thus the calculations can all be done in at worst the maximum of $O(n \log n)$ and $O(t^2)$ for length n and error-correction t. The t^2 is a function of the error-correction capability t. Now in the vast majority of cases, the required distance $2t+1$ satisfies $t \leq \sqrt{(n)}$; in these cases, all the calculations can be done in at worst $O(n \log n)$ calculations. If super fast calculations of the kernel of a Hankel $t \times (t+1)$ matrix are employed as proposed then the calculations can be done in $\max\{O(n \log n), O(t \log^2 t)\}$ operations. This is certainly of $O(n \log n)$ when $\log^2 t \leq n$ or $\log t \leq \sqrt{\log n}$.

References

Blahut, R.E. (2003) *Algebraic Codes for data transmission*, Cambridge University Press, Cambridge, UK.

Chernoff, H. (1952) 'A measure of asymptotic efficiency for tests of a hypothesis based on a sum of observations', *Annals of Math. Stats.*, Vol. 23, No. 4, pp.493–507. See also the many lecture notes on the topic available on-line and elsewhere.

Duursma, I. and Kötter, R. (1994) 'Error-locating pairs for cyclic codes', *IEEE Trans. Inform. Theory*, Vol. 40, pp.1108–1121.

Hurley, P. and Hurley, T. (2009) 'Codes from zero-divisors and units in group rings', *Int. J. Inform. and Coding Theory*, Vol. 1, pp.57–87.

Hurley, P. and Hurley, T. (2007) 'Module codes in group rings', *ISIT2007*, Nice, pp.1981–1985.

Hurley, P. and Hurley, T. (2010a) 'Block codes from matrix and group rings', in Woungang, I., Misra, S. and Misma, S.C. (Eds.): *Selected Topics in Information and Coding Theory*, World Scientific, Chapter 5, Singapore, pp.159–194.

Hurley, P. and Hurley, T. (2010b) 'LDPC and convolutional codes from matrix and group rings', in Woungang, I., Misra, S. and Misma, S.C. (Eds.): *Selected Topics in Information and Coding Theory*, World Scientific, Chapter 6, Singapore, pp.195–237.

Hurley, B. and Hurley, T. (2014) 'Systems of MDS codes from units and idempotents', *Discrete Math.*, Vol. 335, pp.81–91.

Hurley, T., McEvoy, P. and Wenus, J. (2010) 'Algebraic constructions of LDPC codes with no short cycles', *Intl. J. Inform. and Coding Theory*, Vol. 1, No. 3, pp.285–297.

Hurley, T. (2007) *Self-Dual, Dual-Containing and Related Quantum Codes from Group Rings*, arXiv:0711.3983.

Hurley, T. (2009) 'Convolutional codes from units in matrix and group rings', *Inter. J. Pure and Applied Mathematics*, Vol. 50, No. 3, pp.431–463.

Hurley, T. (2014) 'Cryptographic schemes, key exchange, public key', *Int. J. Pure and Applied Maths.*, Vol. 93, No. 6, pp.897–927.

Hurley, T. (2016a) *Convolutional Codes from Unit Schemes*, ArXiv 1412.1695, 22 pp.

Hurley, T. (2016b) 'Group rings and rings of matrices', *Inter. J. Pure and Appl. Math.*, Vol. 31, No. 3, pp.319–335.

Hurley, T. (2017) 'Solving underdetermined systems with error correcting codes', *Intl. J. Information and Coding Theory*, Vol. 4, No. 4, pp.201–221.

McEliece, R.J. (1978) 'A public-key cryptosystem based on algebraic coding theory', *DSN (Deep Space Network) Progress Report 42-44*, pp.114–116.

McEliece, R.J. (2002) *Theory of Information and Coding*, 2nd ed., Cambridge University Press, Cambridge, UK.

Pellikaan, R. (1992) 'On decoding by error location and dependent sets of error positions', *Discrete Math.*, Vols. 106/107, pp.369–381.

Website

'GAP – Groups, Algorithms and Programming', www.gap-system.org

Notes

[1]This section is independent of the succeeding sections.
[2]That $\omega = 7 \mod 29$ is used here is coincidental to the size of the matrix.
[3]Other choices are possible.
[4]Note that 13 is a base 3 repunit.
[5](This general Vandermonde case can be done similar to that of Section 7 of Hurley (2017) although in that paper the field is $\mathbb{C}$.)