ON CONNECTIONS BETWEEN GRAPHS, CODES, QUANTUM STATES, AND BOOLEAN FUNCTIONS

ON CONNECTIONS BETWEEN GRAPHS, CODES, QUANTUM STATES, AND BOOLEAN FUNCTIONS

Lars Eirik Danielsen

DISSERTATION FOR
THE DEGREE OF PHILOSOPHIAE DOCTOR

THE SELMER CENTER
DEPARTMENT OF INFORMATICS
UNIVERSITY OF BERGEN
NORWAY

MAY 2008

CONTENTS

ABSTRACT

We study objects that can be represented as graphs, error-correcting codes, quantum states, or Boolean functions. It is known that self-dual additive codes, which can also be interpreted as quantum states, can be represented as graphs, and that two codes are equivalent when the corresponding graphs are equivalent with respect to local complementation (LC). We give classifications of such codes. Circulant graph codes are introduced, and it is shown that some of these codes have highly regular graph representations. We show that the orbit of a bipartite graph under edge local complementation (ELC) corresponds to the equivalence class of a binary linear code. We classify ELC orbits, give a new method for classifying binary linear codes, and show that the information sets and the minimum distance of a code can be derived from the corresponding ELC orbit. Self-dual additive codes over $\mathrm{GF}(4)$ can also be interpreted as quadratic Boolean functions. In this context we define PAR_{IHN}, peak-to-average power ratio with respect to the $\{I, H, N\}^n$ transform, and prove that PAR_{IHN} equals the size of the maximum independent set over the associated LC orbit of graphs. We define the aperiodic propagation criteria (APC) of a Boolean function, and show that it is related to the minimum distance of a self-dual additive code over $\mathrm{GF}(4)$, and to the degree of entanglement in the associated quantum state. We give a generalization to non-quadratic Boolean functions, and relate APC to known cryptographic criteria. Interlace polynomials encode many properties of the LC and ELC orbits of a graph. We enumerate interlace polynomials and circle graphs, show that there exist non-unimodal interlace polynomials, and relate properties of interlace polynomials to properties of codes and quantum states. We define self-dual bent functions, and provide constructions and classifications.

ACKNOWLEDGEMENTS

The three years I have worked on this thesis have been both interesting and challenging. I wish to use this opportunity to thank all those whose advice and encouragement I have received. Firstly, I would like to express my gratitude to my supervisor, Matthew G. Parker, for his guidance, enthusiasm, and constant supply of ideas. Secondly, I thank my co-supervisor Tor Helleseth and all colleagues at the Selmer Center for providing an excellent work environment. Working in a group that has been described by the Norwegian Research Council as "one of the gems on the Norwegian Scientific scene" has been truly inspiring. Finally, I give my thanks to Kristin and my family for their support.

INTRODUCTION TO THE THESIS

1 MOTIVATION

This thesis is mainly a study of objects that can be represented as graphs, error-correcting codes, quantum states, or Boolean functions. These different ways of viewing what is fundamentally the same object, give different insights to the properties of the object, and give rise to different constructions and generalizations.

Generating orbits of graphs under certain local operations give us efficient methods for classifying codes. We mainly consider codes that have applications in quantum computation, and which can also be represented as quantum graph states. Entangled graph states can be used, for instance, as a resource for measurement-based quantum computation. Graphs have a natural generalization to weighted graphs, which correspond to more general non-binary quantum states. Certain properties of a quantum graph state can be derived from the interlace polynomials of the associated graph.

A graph can also be represented as a quadratic Boolean function. There is a natural generalization to non-quadratic Boolean functions, which can still be interpreted as quantum states. Boolean functions have applications in cryptography, and it is interesting to observe that properties of Boolean functions that are relevant in cryptography are related to measures of entanglement in quantum states.

2 GRAPHS

A *graph* is a pair $G = (V, E)$ where V is a set of *vertices*, and $E \subseteq V \times V$ is a set of *edges*. A graph with n vertices can be represented by an $n \times n$ *adjacency matrix* Γ, where $\gamma_{uv} = 1$ if $\{u, v\} \in E$, and $\gamma_{uv} = 0$ otherwise. We will consider *simple undirected* graphs whose adjacency matrices

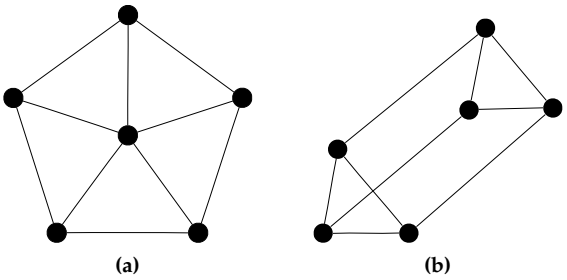


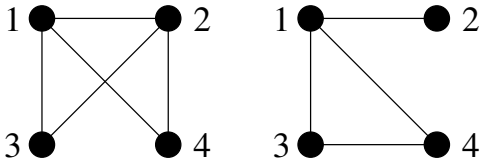
Fig. 1: *Two Graph Representations of the Hexacode*

are symmetric with all diagonal elements being 0. The *neighbourhood* of $v \in V$, denoted $N_v \subset V$, is the set of vertices connected to v by an edge. The number of vertices adjacent to v is called the *degree* of v. The *induced subgraph* of G on $W \subseteq V$ contains vertices W and all edges from E whose endpoints are both in W. The *complement* of G is found by replacing E with $V \times V - E$, i.e., the edges in E are changed to non-edges, and the non-edges to edges.

As an example, consider the graph in Fig. 1a. We will usually consider graphs up to *isomorphism*, i.e., we ignore the labeling of the vertices. Choosing an arbitrary labeling of vertices, the adjacency matrix of this graph is

$$\Gamma = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

A graph operation that is central in this thesis is *local complementation* (LC) [1], which is defined as follows. Given a graph $G = (V, E)$ and a vertex $v \in V$, LC on v transforms G into $G * v$, where we have replaced the induced subgraph of G on N_v by its complement. As an example, we will perform LC on vertex 1 of the graph G, shown in Fig. 2a. We see that the neighbourhood of 1 is $N_1 = \{2, 3, 4\}$ and that the induced subgraph on the neighbourhood has edges $\{2, 3\}$ and $\{2, 4\}$. The complement of this subgraph contains the single edge $\{3, 4\}$. The resulting LC image, $G * 1$, is seen in Fig. 2b. As another example, consider the graph shown in Fig. 1b. An LC operation on any vertex of this graph produces the graph shown in Fig. 1a. An LC operation



(a) *The Graph G* **(b)** *The Graph* $G * 1$

Fig. 2: *Example of Local Complementation*

on the vertex in the centre of the graph shown in Fig. 1a gives the same graph, up to isomorphism. LC operations on any of the other five vertices produces the graph shown in Fig. 1b.

Another operation that we use is *edge local complementation* (ELC) [2], sometimes called the *pivot* operation, which can be defined in terms of LC. Given a graph $G = (V, E)$ and an edge $\{u, v\} \in E$, ELC on $\{u, v\}$ transforms G into $G^{(uv)} = G * u * v * u = G * v * u * v$. The *LC orbit* of a graph G is the set of all graphs that can be obtained by performing any sequence of LC operations on G. As an example, the two graphs shown in Fig. 1 make up a complete LC orbit, up to isomorphism. Similarly, the *ELC orbit* of G comprises all graphs that can be obtained by performing any sequence of ELC operations on G. LC and ELC were originally introduced in the context of *isotropic systems* [2].

Associated with a graph G are *interlace polynomials* $q(G, x)$ [3] and $Q(G, x)$ [4]. The polynomial $q(G)$, whose introduction was motivated by a problem related to DNA sequencing [5], encodes certain properties of the ELC orbit of G and is defined as follows. If G is a graph with n vertices and no edges, $q(G) = x^n$. For any other graph $G = (V, E)$, choose an arbitrary edge $\{u, v\} \in E$, and let $q(G) = q(G \setminus u) + q(G^{(uv)} \setminus u)$, where $G \setminus u$ is the graph G with vertex u and all edges incident on u removed. $Q(G)$ similarly encodes properties of the LC orbit of G and is defined as follows. If G is an edgeless graph with n vertices, $Q(G) = x^n$. For any other graph $G = (V, E)$, choose an arbitrary edge $\{u, v\} \in E$, and let $Q(G) = Q(G \setminus u) + Q(G^{(uv)} \setminus u) + Q(G * u \setminus u)$. Properties of a graph that can be obtained from its interlace polynomials include the number of vertices, the number of connected components, and the number of perfect matchings. Most interesting for our purposes is that the degree of Q equals the size of the largest *independent set* over all members of the LC orbit of G, and that the value of $Q(G, 4)$ equals 2^n times the number of induced *Eulerian* subgraphs of G. An independent

set is a set of vertices such that no pair are connected by an edge, and a graph is Eulerian if all vertices have even degree. As an example, both graphs in Fig. 1 have interlace polynomials $q(G) = 12x + 10x^2$ and $Q(G) = 108x + 45x^2$. The fact that $\deg(Q) = 2$ matches the observation that none of the two graphs have an independent set of size greater than two. That $\frac{Q(G,4)}{2^6} = 18$ means that each graph has 18 Eulerian subgraphs.

3 Codes

Coding theory [6] deals with techniques for detecting and correcting errors in information that has been affected by noise. A binary linear code, $\mathcal{C}$, is a linear subspace of $\mathrm{GF}(2)^n$ of dimension k, where $0 \leq k \leq n$. $\mathcal{C}$ is called an $[n, k]$ code, and the 2^k elements of $\mathcal{C}$ are called *codewords*. Such a code can be utilized by mapping a block of k information bits onto one of the 2^k n-bit codewords. Upon receiving n bits, the receiver can detect errors simply by observing that the bits do not form a valid codeword. He can then perform error-correction, typically by choosing the codeword that is closest to the received bits in terms of *Hamming distance*. The Hamming distance between two vectors is the number of coordinates where they have different values. For error-detection to be successful the number of bit positions in error must be at most $d - 1$, where d is the *minimum distance* of the code. For error-correction to succeed, the number of errors must be at most $\left\lfloor \frac{d-1}{2} \right\rfloor$. The minimum distance of a code is the minimal Hamming distance between any two codewords. A linear code with minimum distance d is called an $[n, k, d]$ code, and can be defined by a $k \times n$ *generator matrix* whose rows span the code. Two codes are considered to be *equivalent* if one can be obtained from the other by some permutation of the coordinates, or equivalently, a permutation of the columns of a generator matrix. A classical problem is to classify codes up to equivalence [7]. We define the *dual* of the code $\mathcal{C}$ as $\mathcal{C}^\perp = \{\boldsymbol{u} \in \mathrm{GF}(2)^n \mid \boldsymbol{u} \cdot \boldsymbol{c} = 0 \text{ for all } \boldsymbol{c} \in \mathcal{C}\}$. $\mathcal{C}$ is called *self-orthogonal* if $\mathcal{C} \subseteq \mathcal{C}^\perp$, and *self-dual* [8] if $\mathcal{C} = \mathcal{C}^\perp$. The classification of self-dual codes is also an active research topic [9].

Besides binary linear codes, in this thesis we study a more special class of codes known as self-dual additive codes over $\mathrm{GF}(4)$ [10, 11], which are of interest due to their connections to quantum codes [12]. We denote the finite field with four elements $\mathrm{GF}(4) = \{0, 1, \omega, \omega^2\}$, where $\omega^2 = \omega + 1$. An $(n, 2^k, d)$ *additive* code over $\mathrm{GF}(4)$ is an additive

subgroup of $\mathrm{GF}(4)^n$ with 2^k codewords of length n and minimum distance d. It can be defined by a $k \times n$ generator matrix. We consider codes that are self-dual with respect to the *Hermitian trace inner product*, $\boldsymbol{u} * \boldsymbol{v} = \sum_{i=1}^{n} (u_i v_i^2 + u_i^2 v_i) \pmod 2$. A self-dual code must be an $(n, 2^n)$ code. Two codes are equivalent if we can get from one to the other by a permutation of coordinates and permutations of the elements $\{1, \omega, \omega^2\}$ in each coordinate.

As an example, consider the $(6, 2^6, 4)$ code, also known as the Hexacode, generated by

$$C = \begin{pmatrix} 1 & 0 & 0 & 1 & \omega & \omega \\ \omega & 0 & 0 & \omega & \omega^2 & \omega^2 \\ 0 & 1 & 0 & \omega & 1 & \omega \\ 0 & \omega & 0 & \omega^2 & \omega & \omega^2 \\ 0 & 0 & 1 & \omega & \omega & 1 \\ 0 & 0 & \omega & \omega^2 & \omega^2 & \omega \end{pmatrix}.$$

It is known that all self-dual additive codes over $\mathrm{GF}(4)$ can be represented as graphs [2, 13–17]. More specifically, every code is equivalent to a code with generator matrix $\Gamma + \omega I$, where Γ is the adjacency matrix of a graph, and I is an identity matrix. For example, the code generated by the matrix C given above is equivalent to the code generated by

$$C' = \Gamma + \omega I = \begin{pmatrix} \omega & 0 & 1 & 0 & 1 & 1 \\ 0 & \omega & 1 & 1 & 0 & 1 \\ 1 & 1 & \omega & 0 & 0 & 1 \\ 0 & 1 & 0 & \omega & 1 & 1 \\ 1 & 0 & 0 & 1 & \omega & 1 \\ 1 & 1 & 1 & 1 & 1 & \omega \end{pmatrix},$$

where Γ is the adjacency matrix of the graph depicted in Fig. 1a. It is also known that if two codes are equivalent, their associated graphs will be related by some sequence of local complementations [2, 16, 17]. This means that the LC orbit of a graph corresponds to the equivalence class of a self-dual additive code over $\mathrm{GF}(4)$. For example, the two graphs shown in Fig. 1 represent the equivalence class of the Hexacode.

More generally, self-dual additive codes can be defined over the finite field $\mathrm{GF}(m^2)$ [18, 19]. These codes can be represented as graphs where the edges are weighted with elements from $\mathrm{GF}(m)$ [14, 15], and a generalization of the LC operation can be used to generate their equivalence classes [20].

4 Quantum States

A quantum computer [21] performs computations by using quantum mechanical systems. The laws of quantum mechanics predict effects, such as *superposition* and *entanglement*, which are very different from the physical reality we ordinarily observe. These effects makes it possible to solve certain problems much more efficiently on a quantum computer than on a classical computer. Shor's algorithm for factoring integers in polynomial time is of particular interest, due to the importance of factoring in public-key cryptography. Quantum computation is an active field of research, although a scalable quantum computer has not yet been built.

Quantum information can be stored in quantum bits, also known as *qubits*. A qubit can be described by a vector $|x\rangle = \binom{\alpha}{\beta} \in \mathbb{C}^2$, where $|\alpha|^2$ is the probability of observing the value 0 when we measure the qubit, and $|\beta|^2$ is the probability of observing 1. If both α and β are non-zero, the qubit has both the value 0 and 1 at the same time, and we call this a *superposition*. Once we have measured the qubit, however, the superposition collapses, and we are left with a classical state that is either 0 or 1 with certainty. A state of n qubits is represented by a normalized complex vector with 2^n elements. The state is called *entangled* if the measurement outcomes of the qubits are not independent. An example of an entangled state of two qubits is the *EPR pair*, $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, with vector representation $(\frac{1}{\sqrt{2}}, 0, 0, \frac{1}{\sqrt{2}})^T$. Once we measure one of these two qubits, we know that the other qubit must have the same value.

Measurement of a quantum state collapses superpositions and destroys entanglement, but we can perform an operation that does not disturb the state. Such a transformation must be represented by a *unitary* matrix, i.e., $UU^\dagger = I$, where $\dagger$ means conjugate transpose. A transformation that acts independently on each qubit of a multi-qubit state is called *local unitary*, and can be written as a *tensor product* (or *Kronecker product*) of 2×2 matrices. We define the tensor product as

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots \\ a_{12}B & a_{22}B & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}.$$

Some kind of error-correction is essential in any model for quantum computation, since an unprotected quantum state will quickly *decohere*,

i.e., be destroyed by interaction with the environment. It is remarkable that quantum error correction is possible [22], since quantum information can not be copied or observed without destroying it, and since quantum errors are continuous. Assuming that errors act independently on each qubit, every possible error can be described as a local unitary transform. Any 2×2 unitary matrix can be written as a linear combination of the *Pauli matrices*,

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad Y = iXZ \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

By using an error correction process that collapses any error operator to a tensor product of Pauli matrices, a quantum code only needs to consider the three possible errors X, Z, and Y. A class of quantum codes that exploit this fact are the *stabilizer codes* [23]. An $[[n, k]]$ stabilizer code encodes k qubits using n qubits, and is an Abelian group generated by a set of $n - k$ commuting Pauli operators. An error is detected by measuring the eigenvalues of these operators. If a state is a valid codeword that has not been affected by error, we will observe the eigenvalue $+1$ for all operators. If there is a detectable error, some eigenvalues would be -1, due to the fact that Pauli matrices anti-commute. In this thesis, we consider $[[n, 0]]$ codes. An $[[n, 0]]$ code is generated by a set of n Pauli operators and represents a single quantum state known as a *stabilizer state*. The minimum distance, d, of an $[[n, 0, d]]$ code is the minimum weight of all error operators in the stabilizer, where the weight of an error operator is the number of tensor components that are different from I. If the minimum distance is high, the stabilizer state is highly entangled.

As an example, consider the $[[6, 0, 4]]$ stabilizer state, generated by the operators

$$\begin{aligned}
&Z \otimes I \otimes I \otimes Z \otimes X \otimes X \\
&X \otimes I \otimes I \otimes X \otimes Y \otimes Y \\
&I \otimes Z \otimes I \otimes X \otimes Z \otimes X \\
&I \otimes X \otimes I \otimes Y \otimes X \otimes Y \\
&I \otimes I \otimes Z \otimes X \otimes X \otimes Z \\
&I \otimes I \otimes X \otimes Y \otimes Y \otimes X.
\end{aligned}$$

We can represent an n-fold tensor product of Pauli matrices as a vector in $\mathrm{GF}(4)^n$, by the mappings $I \mapsto 0$, $Z \mapsto 1$, $X \mapsto \omega$, and $Y \mapsto \omega^2$.

In this way, it is possible to represent stabilizer codes as codes over GF(4) [12]. That the stabilizer is Abelian and that all operators commute means that the code over GF(4) must be additive and self-orthogonal. Stabilizer states correspond to self-dual additive codes over GF(4). As an example, the $[[6,0,4]]$ code given above corresponds to the self-dual additive $(6,2^6,4)$ Hexacode. Since all self-dual additive codes over GF(4) can be represented as graphs, this also holds for all stabilizer states. Thus the $[[6,0,4]]$ code corresponds to the LC orbit shown in Fig. 1. A stabilizer state represented in graph form is called a *graph state* [24]. Graph states can be used as a resource for measurement-based quantum computation [25], where carefully chosen measurements on an entangled state are used to execute a quantum algorithm.

Instead of qubits, we could consider the more general m-level *qudits*, and the corresponding non-binary stabilizer codes [18, 19].

5 Boolean Functions

A Boolean function of n variables is a function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$. It can be represented as a vector in $\mathbb{Z}_2^{2^n}$ listing the values of $f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{Z}_2^n$, called a *truth table*. It can also be represented in *algebraic normal form*, as a sum of monomials of n variables. The *degree* of a Boolean function is the degree of the highest degree term in its algebraic normal form. A graph on n vertices can be represented as a quadratic Boolean function of n variables, by simply mapping the edge $\{u, v\}$ to the monomial $x_u x_v$. Furthermore, given a Boolean function $f(\boldsymbol{x})$ of n variables, the vector $\boldsymbol{s} = 2^{-\frac{n}{2}}(-1)^{f(\boldsymbol{x})}$ can be interpreted as the state vector of a quantum state of n qubits. This is a convenient way of mapping a graph to its corresponding graph state, but is also valid for Boolean functions of degree higher than two.

Local unitary transforms of the vector $\boldsymbol{s}$ preserve the entanglement of the associated quantum state. We are particularly interested in the set of transforms

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad N = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix},$$

where $i^2 = -1$. These three matrices are called the identity, Hadamard, and Negahadamard kernels, and are generators for the *local Clifford group*. The $\{I, H, N\}^n$ set of transforms is given by the 3^n n-fold tensor combinations of I, H, and N. Applied to the vector $\boldsymbol{s}$ these transforms

produce 3^n spectra. A transform of $\boldsymbol{s}$ that, to within normalization, contains only entries ± 1 is a *flat* spectrum and can be interpreted as $2^{-\frac{n}{2}}(-1)^{f'(\boldsymbol{x})}$, where f' is a Boolean function we consider to be equivalent to f. It has been shown that the set of flat $\{I, H, N\}^n$ spectra of a quadratic Boolean function corresponds to the LC orbit of the associated graph [26]. In fact, LC on a vertex v corresponds to the application of N to the variable x_v, and ELC on an edge $\{u, v\}$ corresponds to the application of H to variables x_u and x_v.

The $\{I, H, N\}^n$ transforms of a Boolean function can tell us something about the degree of entanglement in the corresponding quantum state [27]. We define PAR_{IHN}, the peak-to-average power ratio with respect to $\{I, H, N\}^n$, as

$$\mathrm{PAR}_{IHN}(\boldsymbol{s}) = 2^n \max_{\substack{U \in \{I,H,N\}^n \\ \boldsymbol{k} \in \mathbb{Z}_2^n}} |S_{\boldsymbol{k}}|^2, \quad \text{where } \boldsymbol{S} = U\boldsymbol{s}.$$

PAR_{IHN} will be a value between 1 and 2^n, and states with a high degree of entanglement will have a low value. More generally, we can define PAR_U with respect to the infinite set of all local unitary transforms, on which PAR_{IHN} is only a lower bound. Since the values $|S_{\boldsymbol{k}}|^2$ are the probabilities of observing each of 2^n basis states associated with $\boldsymbol{S}$, and a local unitary transformation corresponds to a change of measurement basis, PAR_U answers the question: If we can use any local measurement basis, what is the highest probability that can be achieved for any basis state? A more well-known measure of entanglement is the *Schmidt measure* [24], which answers the question: For all local unitary transforms $\boldsymbol{S} = U\boldsymbol{s}$, what is the lowest number of non-zero coefficients in $\boldsymbol{S}$? Bounds on the Schmidt measure can be obtained from PAR_U and PAR_{IHN} [28]. It has also been shown [29] that PAR_U is equivalent to the *geometric measure* [30] of entanglement for a pure quantum state, and that PAR_{IHN} is an upper bound on the geometric measure. Another property related to entanglement is the *Clifford merit factor* (CMF) [31],

$$\mathrm{CMF}(\boldsymbol{s}) = \frac{6^n}{\sum_{\substack{U \in \{I,H,N\}^n \\ \boldsymbol{k} \in \mathbb{Z}_2^n}} |S_{\boldsymbol{k}}|^4 - 6^n}, \quad \text{where } \boldsymbol{S} = U\boldsymbol{s}.$$

It can be shown that CMF is invariant under any local unitary transform [31]. As an example, the graph shown in Fig. 1a corresponds to the Boolean function $f(\boldsymbol{x}) = x_1x_3 + x_1x_5 + x_1x_6 + x_2x_3 + x_2x_4 + x_2x_6 + x_3x_6 + x_4x_5 + x_4x_6 + x_5x_6$, which has $\mathrm{PAR}_{IHN} = 4$ and $\mathrm{CMF} \approx 1.7234$,

which are known to be the lowest values for any quadratic or cubic function of 6 variables [28].

Boolean functions have important applications in cryptography, for instance as components of S-boxes in block ciphers, and as filtering functions in stream ciphers. An interesting class of functions in this context are the *bent* [32] functions, due to their perfect *non-linearity*. A Boolean function is called bent if its *Walsh-Hadamard transform*, defined by the unitary matrix $H^{\otimes n}$, is flat. The Walsh-Hadamard transform of a bent function f can be interpreted as the truth table of another Boolean function $\tilde{f}$, called the *dual* of f. Note that the transform $H^{\otimes n}$ is one of the 3^n $\{I, H, N\}^n$ transforms. There are many properties that Boolean functions used in cryptography should satisfy. For instance, to be secure against differential cryptanalysis, the component Boolean functions of an S-box should satisfy the *propagation criteria* (PC) and the *extended propagation criteria* (EPC) [33], both derived from the periodic *autocorrelation* spectrum of the function.

6 Summary of Papers

The thesis consists of seven papers. A short overview of each paper follows.

6.1 Paper I

The first paper, entitled "On the classification of all self-dual additive codes over GF(4) of length up to 12," is co-authored with Matthew G. Parker and was published in *J. Combin. Theory Ser. A* 113(7), 1351–1367, 2006. An earlier version of the paper was also presented at the Fourth International Workshop on Optimal Codes and Related Topics (OC 2005) in Pamporovo, Bulgaria.

We give an introduction to self-dual additive codes over GF(4) and their graph representation. By recursively applying LC operations and checking for graph isomorphism, we can generate the entire LC orbit of a graph, which corresponds to the equivalence class of a self-dual additive code over GF(4). Self-dual additive codes over GF(4) of length up to 9 have previously been classified, but by running our graph-based algorithm on a parallel cluster computer, we are able to extend the classification to length 12, which is the main result of this paper. By exploiting the fact that codes of Type II correspond to anti-Eulerian graphs, where all vertices have odd degree, we are also able

to classify all extremal Type II codes of length 14. We also solve an open problem by finding that the smallest Type I and Type II codes with trivial automorphism group have lengths 9 and 12, respectively.

6.2 Paper II

The second paper is entitled "Graph-based classification of self-dual additive codes over finite fields," and is a preprint of a paper that has been submitted for publication.

This is a generalization of the work in Paper I, in that we consider self-dual additive codes over GF(m^2). It is known that these codes correspond to quantum stabilizer states over GF(m), that they can be represented as weighted graphs, and that a generalization of the LC operation generates the equivalence class of a code. By using a graph-based algorithm, similar to the one used in Paper I, all self-dual additive codes over GF(9), GF(16), and GF(25) up to lengths 8, 6, and 6, respectively, are classified. By using an extension technique, we are also able to classify all extremal self-dual additive codes over GF(9) of length 9 and 10. Assuming that the MDS conjecture holds, this means that all self-dual additive MDS codes over GF(9) are classified. We show that the known mass formula for self-dual additive codes over GF(4) is easily generalized to GF(m^2), and use this to derive bounds on the number of inequivalent codes. We also prove that the minimum distance of a code is equal to one plus the minimum vertex degree over all graphs in the associated generalized LC orbit. We introduce the concept of a circulant graph code, and perform a computer search which reveals this class to contain many strong codes. It is mentioned that some of the circulant graph codes correspond to highly regular graph structures, in particular the "nested clique" graphs. This concept is explored further in Papers IV and VI.

6.3 Paper III

The third paper, entitled "Edge local complementation and equivalence of binary linear codes," is co-authored with Matthew G. Parker. It has been accepted for publication in *Des. Codes Cryptogr.*, after being presented at the International Workshop on Coding and Cryptography (WCC 2007) in Versailles, France.

Whereas Papers I and II studied self-dual additive codes, interesting for their applications in quantum error-correction, this paper deals

with classical binary linear codes. Like a self-dual additive code can be represented as a graph, it is common to represent a binary linear code as a bipartite graph. Our contribution is to prove that the equivalence class of the code can be generated by recursively applying edge local complementation (ELC) on the associated bipartite graph. Although the improvement of our graph-based approach over classical classification algorithms is not as great for binary linear codes as it was for self-dual additive codes, we claim that our algorithm has running time comparable to the best known algorithm. With our algorithm it is possible to classify all binary linear codes of length at least up to 15. The classification of all codes of length greater than 14 has been presented as an open problem. With a minor modification to the ELC operation, we obtain an algorithm for finding all information sets of a code. We also explain the relationship between the number of ELC orbits of bipartite graphs on n vertices and the number of inequivalent binary linear codes of length n. We prove that the minimum distance of a binary linear code is equal to one plus the minimum vertex degree over all graphs in the associated ELC orbit. We also classify the ELC orbits of all graphs on up to 12 vertices. Although not applicable to binary linear codes, the ELC orbits of non-bipartite graphs may be of interest in the context of quantum graph states. They are also useful in the study of interlace polynomials, as will be seen in Paper VI.

6.4 Paper IV

The fourth paper is entitled "Spectral orbits and peak-to-average power ratio of Boolean functions with respect to the $\{I, H, N\}^n$ transform," and is a joint work with Matthew G. Parker. It was presented at Sequences and Their Applications (SETA 2004) in Seoul, South Korea, and published in *Lecture Notes in Comput. Sci.* 3486, 373–388, 2005.

The classification of all self-dual additive codes over GF(4) of length up to 12 was first reported in this paper, albeit in a much more condensed version than in Paper I. It is also shown that certain self-dual additive codes over GF(4) with high minimum distance can be represented by graphs with a highly regular "nested clique"-structure. The connections between self-dual additive codes, graphs, quantum codes, and stabilizer states have already been elaborated on. In this paper the link to Boolean functions is made. We explain how a graph can be represented as a quadratic Boolean function, and that the LC orbit of the graph can be obtained as the set of flat $\{I, H, N\}^n$ trans-

forms of the Boolean function. PAR_{IHN} is defined, and it is shown that it is correlated with the minimum distance of the corresponding self-dual additive code, and with the degree of entanglement in the corresponding stabilizer state. We prove that PAR_{IHN} of a quadratic Boolean function is given by the size of the largest independent set over all graphs in the associated LC orbit, and give bounds on this value derived from Ramsey numbers and from computational results. The $\{I, H, N\}^n$ transform and PAR_{IHN} remain well-defined also for Boolean functions with degree higher than two. These functions do not correspond to stabilizer states, but can still be interpreted as quantum states. Boolean functions of high degree and strong spectral properties are also of interest in cryptography. We propose a construction technique to generate non-quadratic Boolean functions with low PAR_{IHN}. The study of Boolean functions related to quantum states and cryptography is continued in Paper V. The properties of the interlace polynomial of a graph are used to prove a theorem in this paper. Interlace polynomials are studied further and some of the results in this paper are extended in Paper VI.

6.5 Paper V

The fifth paper is a joint work with T. Aaron Gulliver and Matthew G. Parker. It is entitled "Aperiodic propagation criteria for Boolean functions," and was published in *Inform. and Comput.* 204(5), 741–770, 2006.

Boolean functions used as components of S-boxes in block ciphers should satisfy the propagation criteria (PC) and the extended propagation criteria (EPC), which are both derived from the periodic autocorrelation spectrum of the function. In this paper, we introduce the aperiodic propagation criteria (APC), derived from the aperiodic autocorrelation spectrum of a Boolean function, and related to the first derivative of the function. We set up a cryptographic scenario where APC is relevant, and define the APC distance of a Boolean function to be the minimum number of plaintext bits that an attacker must both know and be able to modify in order to succeed with a differential attack. Surprisingly, the APC distance of a quadratic Boolean function is equal to the minimum distance of the corresponding self-dual additive code over $\mathrm{GF}(4)$. This means that Boolean functions with high APC distance also correspond to highly entangled graph states, and provides a link between quantum entanglement and cryptographic

criteria. The interpretation of Boolean functions as quantum states is also extended to non-quadratic functions, which can be represented as hypergraphs. Non-quadratic functions with high APC distance could be of interest in both cryptography and quantum error-correction. We give examples of cubic functions of 7 and 8 variables that have APC distance equal to the best quadratic functions of the same number of variables. Just as LC operations preserve the minimum distance of the associated self-dual additive code over GF(4), it is proved that the $\{I, H, N\}^n$ set of transforms preserve the APC distance of any Boolean function. It is explained how the $\{I, H, N\}^n$ transform can be used to generate orbits of equivalent Boolean functions, a technique that was also used in Paper IV. The orbits of non-quadratic functions can be viewed as generalized LC orbits. Finally, we perform a differential analysis of a few state-of-the-art S-boxes with respect to both periodic and aperiodic autocorrelation. High aperiodic biases are found, but it might be difficult to exploit these biases in a practical attack.

6.6 Paper VI

The sixth paper, a preprint entitled "Interlace polynomials: Classification, unimodality, and connections to codes," is co-authored with Matthew G. Parker.

In this paper, we study the properties of the interlace polynomials $q(G)$ and $Q(G)$ associated with a graph G, and the connections to the self-dual additive code and quantum state that G represents. It is known that the minimum distance of a self-dual additive code over GF(4) is equal to one plus the minimum vertex degree over all graphs in the associated LC orbit. This value can not be obtained from any interlace polynomial, but we show that other values that are correlated with the minimum distance are encoded in the interlace polynomial Q. In Paper IV, we proved that PAR_{IHN} of a quadratic Boolean function is given by the size of the largest independent set over all graphs in the associated LC orbit. This value can also be obtained as the degree of $Q(G)$. It has also been shown that from the value $Q(G, 4)$, which gives us the number of induced Eulerian subgraphs of G, we can obtain the Clifford merit factor (CMF). Our computational results show that the interlace polynomial Q of a graph that corresponds to a self-dual additive code over GF(4) with high minimum distance, will also have low degree and a low value of $Q(G, 4)$. In particular, we have not found a single example where the graph of order n with lowest known $Q(G, 4)$

does not also correspond to a code with the highest known minimum distance.

Graphs that are not LC equivalent may still have the same interlace polynomial. It is interesting to know how many distinct interlace polynomials exist, and we give an enumeration of the interlace polynomials q and Q of all graphs of order up to 12.

A graph is called a circle graph if its vertices can be represented as chords on a circle, such that every edge corresponds to a intersection of two chords. Previously, all circle graphs of order up to 9 had been enumerated. We give an enumeration of all circle graphs of order up to 12. When it comes to the application of graphs as self-dual additive codes over GF(4) we show that circle graphs, as well as bipartite graphs, are not well suited. We also revisit the "nested clique"-graphs, and give some indications as to why these graphs are particularly good.

A sequence is unimodal if it is non-decreasing up to some coefficient, and the rest of the sequence is non-increasing. It has been conjectured that all interlace polynomials q have unimodal coefficient sequences. We show that there exist graphs of order 10 with interlace polynomials q whose coefficient sequences are non-unimodal, thereby disproving this conjecture. We also show that by various extension techniques, we can find graphs of any order greater than 10 with non-unimodal interlace polynomials. With these extension techniques, we can obtain all graphs of order 11 with non-unimodal interlace polynomials, and all but four graphs of order 12. We verify that for graphs of order up to 12, all interlace polynomials Q have unimodal coefficients, and we conjecture that this holds for all Q.

6.7 Paper VII

The seventh and last paper is a joint work with Claude Carlet, Matthew G. Parker, and Patrick Solé. It is entitled "Self-dual bent functions," and will be presented by Patrick Solé at the Fourth International Workshop on Boolean Functions: Cryptography and Applications (BFCA 2008) in Copenhagen, Denmark.

We introduce self-dual bent functions as the subset of bent functions where $\tilde{f} = f$, where $\tilde{f}$ is the dual of f. We call a bent function anti-self-dual if $\tilde{f}$ is the complement of f. A spectral characterization in terms of the Rayleigh quotient of $H^{\otimes n}$ allows us to give a simple and efficient search algorithm which makes it possible to classify all self-dual bent function of up to six variables and all quadratic self-dual bent functions

of 8 variables. We also derive primary and secondary constructions for self-dual bent functions, and describe the operations on Boolean functions that preserve self-duality. If the function is a quadratic form, ELC on the corresponding graph is one such operation. More generally, $\{I, H\}^n$ transforms, the action of the orthogonal group, and a restricted set of affine offsets preserve self-duality. We have classified self-dual bent functions with respect to the latter two of these symmetries.

There are several connections between the topic of this paper and the previous six papers. The transform $H^{\otimes n}$ is one of the 3^n $\{I, H, N\}^n$ transforms studied in Papers IV and V. Finding Boolean functions that have good properties with respect to subsets of the $\{I, H, N\}^n$ transforms may provide clues as to what functions are optimal with respect to the complete set of $\{I, H, N\}^n$ transforms. There is also a relationship between self-dual bent functions and self-dual codes. We show that quadratic self-dual bent functions obtained from the Maiorana-MacFarland construction correspond to parity check matrices of self-dual binary codes.

7 Open Problems

This thesis deals with a very diverse topic and presents many open problems and areas for further research. Some of the open problems are listed here.

- In Papers I, II, and III, we classify different types of codes. It would of course be of interest to improve the classification algorithms and extend these classifications.
- In Paper II, we give an algorithm for classifying binary linear codes. There are perhaps some special classes of binary linear codes where our algorithm is particularly effective.
- Analogous to the generalization from Paper I to Paper II, it should be possible to generalize the results in Paper III, i.e., devise an algorithm for classifying non-binary linear codes, by defining a generalized ELC operation for weighted graphs.
- We have only studied self-dual additive codes, corresponding to $[[n, 0, d]]$ stabilizer codes. It would be interesting to also consider self-orthogonal additive codes, or $[[n, k, d]]$ stabilizer codes. These codes are known to have graph representations, but a graph

operation similar to LC that generates the equivalence class of a code is not known.

- For lengths between 23 and 27, the optimal minimum distance of a self-dual additive code over GF(4) is not known. In particular, the existence of a $(24, 2^{24}, 10)$ code remains an open question.

- In Paper II, we generate generalized LC orbits of graphs where the edges are weighted with elements from GF(m), but a generalization of LC to graphs with weights from an Abelian group that is not a finite field is not known.

- The "nested clique" graphs, studied in Papers II, IV, and VI, might lead to a more formal construction, yielding codes with a predictable minimum distance. For self-dual additive codes over GF(4) with minimum distance greater than 8, we have not found any "nested clique" representations, nor any graph representation where all vertices have minimal degree, i.e., one less than the minimum distance. Perhaps a more general regular graph structure than the "nested clique" is required for higher minimum distances.

- In Paper II we introduce circulant graph codes. For self-dual additive codes over GF(4) of length up to 30, the best codes produced by the circulant graph construction have exactly the same minimum distance as codes obtained by the more general circulant code construction. It would be interesting to find conditions for when a circulant code is equivalent to a circulant graph code.

- In Papers IV and VI we give bounds on PAR_{IHN} and CMF from computational results. An open problem is to provide more formal bounds on these values.

- Is it possible to give a bound on the size of the largest independent set in the LC orbit of a Paley graph, or any other family of graphs?

- Finding a new construction technique for graphs or codes with good properties (PAR_{IHN}, CMF, or minimum distance) would be an important result.

- Construction techniques for non-quadratic Boolean functions with good properties (PAR_{IHN}, CMF, or APC distance) are also of interest. In particular, we would like to find a non-quadratic

function with better properties than the best quadratic function of the same number of variables.

- It is not obvious if and how a non-quadratic Boolean function can be employed as a quantum error-correcting code.
- It is known that bipartite graphs do not yield strong self-dual additive codes, and in Paper IV, we see that circle graphs are not good either. Is it possible to give bounds on the properties of a circle graph, and are there other classes of graphs that should be avoided?
- In Papers IV and VI, we calculate the PAR_{IHN} of graphs. It would be interesting to also be able to calculate PAR_U with respect to the infinite set of all unitary transforms. For LC orbits that contain bipartite graphs, this value is equal to PAR_{IHN}, but in general we can only find lower bounds.
- Interlace polynomials are explored in Paper VI, and in Paper II we see that there is a generalized LC operation for weighted graphs. Does this mean that generalized interlace polynomials for weighted graphs can also be defined?
- The aperiodic propagation criteria (APC) is introduced in Paper V, but a practical application of aperiodic autocorrelation to the cryptanalysis of symmetric primitives has not yet been found.
- Another problem is to extend the classification of self-dual bent functions from Paper VII.
- We have seen that certain self-dual bent functions correspond to self-dual binary codes. Finding more connections to the theory of self-dual codes is a topic for further research.

References

[1] de Fraysseix, H.: Local complementation and interlacement graphs. *Discrete Math.* 33(1), 29–35, 1981.

[2] Bouchet, A.: Graphic presentations of isotropic systems. *J. Combin. Theory Ser. B* 45(1), 58–76, 1988.

[3] Arratia, R., Bollobás, B., Sorkin, G. B.: The interlace polynomial of a graph. *J. Combin. Theory Ser. B* 92(2), 199–233, 2004. arXiv:math.CO/0209045.

[4] Aigner, M., van der Holst, H.: Interlace polynomials. *Linear Algebra Appl.* 377, 11–30, 2004.

[5] Arratia, R., Bollobás, B., Coppersmith, D., Sorkin, G. B.: Euler circuits and DNA sequencing by hybridization. *Discrete Appl. Math.* 104(1–3), 63–96, 2000.

[6] MacWilliams, F. J., Sloane, N. J. A.: *The Theory of Error-Correcting Codes*. Elsevier, New York, 1978.

[7] Kaski, P., Östergård, P. R. J.: *Classification Algorithms for Codes and Designs*, *Algorithms and Computation in Mathematics*, vol. 15. Springer, Berlin, 2006.

[8] Rains, E. M., Sloane, N. J. A.: Self-dual codes. In *Handbook of Coding Theory*, pp. 177–294. North-Holland, Amsterdam, 1998. arXiv:math.CO/0208001.

[9] Huffman, W. C.: On the classification and enumeration of self-dual codes. *Finite Fields Appl.* 11(3), 451–490, 2005.

[10] Höhn, G.: Self-dual codes over the Kleinian four group. *Math. Ann.* 327(2), 227–255, 2003. arXiv:math.CO/0005266.

[11] Gaborit, P., Huffman, W. C., Kim, J.-L., Pless, V.: On additive GF(4) codes. In *Codes and Association Schemes*, *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, vol. 56, pp. 135–149. Amer. Math. Soc., Providence, RI, 2001.

[12] Calderbank, A. R., Rains, E. M., Shor, P. M., Sloane, N. J. A.: Quantum error correction via codes over GF(4). *IEEE Trans. Inform. Theory* 44(4), 1369–1387, 1998. arXiv:quant-ph/9608006.

[13] Schlingemann, D.: Stabilizer codes can be realized as graph codes. *Quantum Inf. Comput.* 2(4), 307–323, 2002. arXiv:quant-ph/0111080.

[14] Schlingemann, D., Werner, R. F.: Quantum error-correcting codes associated with graphs. *Phys. Rev. A* 65(1), 2002. arXiv:quant-ph/0012111.

[15] Grassl, M., Klappenecker, A., Rötteler, M.: Graphs, quadratic forms, and quantum codes. In *Proc. IEEE Int. Symp. Inform. Theory*, p. 45. 2002. arXiv:quant-ph/0703112.

[16] Glynn, D. G., Gulliver, T. A., Maks, J. G., Gupta, M. K.: The geometry of additive quantum codes, 2004. Submitted to Springer.

[17] Van den Nest, M., Dehaene, J., De Moor, B.: Graphical description of the action of local Clifford transformations on graph states. *Phys. Rev. A* 69(2), 2004. arXiv:quant-ph/0308151.

[18] Rains, E. M.: Nonbinary quantum codes. *IEEE Trans. Inform. Theory* 45(6), 1827–1832, 1999. arXiv:quant-ph/9703048.

[19] Ketkar, A., Klappenecker, A., Kumar, S., Sarvepalli, P. K.: Nonbinary stabilizer codes over finite fields. *IEEE Trans. Inform. Theory* 52(11), 4892–4914, 2006. arXiv:quant-ph/0508070.

[20] Bahramgiri, M., Beigi, S.: Graph states under the action of local Clifford group in non-binary case, 2006. Preprint. arXiv:quant-ph/0610267.

[21] Nielsen, M. A., Chuang, I. L.: *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2000.

[22] Shor, P. W.: Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A* 52(4), 1995.

[23] Gottesman, D.: *Stabilizer Codes and Quantum Error Correction*. Ph.D. thesis, Caltech, May 1997. arXiv:quant-ph/9705052.

[24] Hein, M., Eisert, J., Briegel, H. J.: Multi-party entanglement in graph states. *Phys. Rev. A* 69(6), 2004. arXiv:quant-ph/0307130.

[25] Raussendorf, R., Browne, D. E., Briegel, H. J.: Measurement-based quantum computation on cluster states. *Phys. Rev. A* 68(2), 2003. arXiv:quant-ph/0301052.

[26] Riera, C., Parker, M. G.: Generalised bent criteria for Boolean functions (I). *IEEE Trans. Inform. Theory* 52(9), 4142–4159, 2006. arXiv:cs.IT/0502049.

[27] Parker, M. G., Rijmen, V.: The quantum entanglement of binary and bipolar sequences. In *Sequences and Their Applications – SETA 2001*, Discrete Math. Theor. Comput. Sci., pp. 296–309. Springer, London, 2002. arXiv:quant-ph/0107106.

[28] Danielsen, L. E.: *On Self-Dual Quantum Codes, Graphs, and Boolean Functions*. Master's thesis, Dept. Informat., Univ. Bergen, Norway, Mar. 2005. arXiv:quant-ph/0503236.

[29] Riera, C., Parker, M. G.: One and two-variable interlace polynomials: A spectral interpretation. In *Coding and Cryptography*, *Lecture Notes in Comput. Sci.*, vol. 3969, pp. 397–411. Springer, Berlin, 2006. arXiv:cs.IT/0504102.

[30] Wei, T.-C., Goldbart, P. M.: Geometric measure of entanglement and applications to bipartite and multipartite quantum states. *Phys. Rev. A* 68(4), 2003. arXiv:quant-ph/0307219.

[31] Parker, M. G.: Univariate and multivariate merit factors. In *Sequences and Their Applications – SETA 2004*, *Lecture Notes in Comput. Sci.*, vol. 3486, pp. 72–100. Springer, Berlin, 2005.

[32] Rothaus, O. S.: On "bent" functions. *J. Combin. Theory Ser. A* 20(3), 300–305, 1976.

[33] Preneel, B., Van Leekwijck, W., Van Linden, L., Govaerts, R., Vandewalle, J.: Propagation characteristics of Boolean functions. In *Advances in Cryptology – EUROCRYPT '90*, *Lecture Notes in Comput. Sci.*, vol. 473, pp. 161–173. Springer, Berlin, 1991.

PAPER I

ON THE CLASSIFICATION OF ALL SELF-DUAL ADDITIVE CODES OVER GF(4) OF LENGTH UP TO 12

Lars Eirik Danielsen Matthew G. Parker

ON THE CLASSIFICATION OF ALL SELF-DUAL ADDITIVE CODES OVER GF(4) OF LENGTH UP TO 12

Lars Eirik Danielsen* Matthew G. Parker*

We consider additive codes over GF(4) that are self-dual with respect to the Hermitian trace inner product. Such codes have a well-known interpretation as quantum codes and correspond to isotropic systems. It has also been shown that these codes can be represented as graphs, and that two codes are equivalent if and only if the corresponding graphs are equivalent with respect to local complementation and graph isomorphism. We use these facts to classify all codes of length up to 12, where previously only all codes of length up to 9 were known. We also classify all extremal Type II codes of length 14. Finally, we find that the smallest Type I and Type II codes with trivial automorphism group have length 9 and 12, respectively.

1 INTRODUCTION

An *additive* code, $\mathcal{C}$, over GF(4) of *length* n is an additive subgroup of $\mathrm{GF}(4)^n$. $\mathcal{C}$ contains 2^k codewords for some $0 \leq k \leq 2n$, and can be defined by a $k \times n$ *generator matrix*, with entries from GF(4), whose rows span $\mathcal{C}$ additively. $\mathcal{C}$ is called an $(n, 2^k)$ code. We denote $\mathrm{GF}(4) = \{0, 1, \omega, \omega^2\}$, where $\omega^2 = \omega + 1$. *Conjugation* of $x \in \mathrm{GF}(4)$ is defined by $\overline{x} = x^2$. The *trace map*, $\mathrm{Tr} : \mathrm{GF}(4) \rightarrow \mathrm{GF}(2)$, is defined by $\mathrm{Tr}(x) = x + \overline{x}$.

*Department of Informatics, University of Bergen, PB 7803, N-5020 Bergen, Norway.

The *Hermitian trace inner product* of two vectors over GF(4) of length n, $\boldsymbol{u} = (u_1, u_2, \ldots, u_n)$ and $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$, is given by

$$\boldsymbol{u} * \boldsymbol{v} = \mathrm{Tr}(\boldsymbol{u} \cdot \overline{\boldsymbol{v}}) = \sum_{i=1}^{n} \mathrm{Tr}(u_i \overline{v_i}) = \sum_{i=1}^{n} (u_i v_i^2 + u_i^2 v_i) \pmod 2. \quad (1)$$

Note that $\boldsymbol{u} * \boldsymbol{v}$ is also the number (modulo 2) of places where $\boldsymbol{u}$ and $\boldsymbol{v}$ have different non-zero values. We define the *dual* of the code $\mathcal{C}$ with respect to the Hermitian trace inner product, $\mathcal{C}^\perp = \{\boldsymbol{u} \in \mathrm{GF}(4)^n \mid \boldsymbol{u} * \boldsymbol{c} = 0 \text{ for all } \boldsymbol{c} \in \mathcal{C}\}$. $\mathcal{C}$ is *self-orthogonal* if $\mathcal{C} \subseteq \mathcal{C}^\perp$. It has been shown that self-orthogonal additive codes over GF(4) can be used to represent *quantum error-correcting codes* [1]. If $\mathcal{C} = \mathcal{C}^\perp$, then $\mathcal{C}$ is *self-dual* and must be an $(n, 2^n)$ code. Self-dual additive codes over GF(4) correspond to zero-dimensional quantum codes, which represent single quantum states. If the code has high minimum distance, the corresponding quantum state is highly *entangled*.

The *Hamming weight* of $\boldsymbol{u}$, denoted $\mathrm{wt}(\boldsymbol{u})$, is the number of nonzero components of $\boldsymbol{u}$. The *Hamming distance* between $\boldsymbol{u}$ and $\boldsymbol{v}$ is $\mathrm{wt}(\boldsymbol{u} - \boldsymbol{v})$. The *minimum distance* of the code $\mathcal{C}$ is the minimal Hamming distance between any two distinct codewords of $\mathcal{C}$. Since $\mathcal{C}$ is an additive code, the minimum distance is also given by the smallest nonzero weight of any codeword in $\mathcal{C}$. A code with minimum distance d is called an $(n, 2^k, d)$ code. The *weight distribution* of the code $\mathcal{C}$ is the sequence $(A_0, A_1, \ldots, A_n)$, where A_i is the number of codewords of weight i. The *weight enumerator* of $\mathcal{C}$ is the polynomial

$$W(x, y) = \sum_{i=0}^{n} A_i x^{n-i} y^i \quad (2)$$

We distinguish between two types of self-dual additive codes over GF(4). A code is of *Type II* if all codewords have even weight, otherwise it is of *Type I*. It can be shown that a Type II code must have even length. Bounds on the minimum distance of self-dual codes were given by Rains and Sloane [2, Theorem 33]. Let d_{I} be the minimum distance of a Type I code of length n. Then d_{I} is upper-bounded by

$$d_{\mathrm{I}} \leq \begin{cases} 2\left\lfloor \frac{n}{6} \right\rfloor + 1, & \text{if } n \equiv 0 \pmod 6 \\ 2\left\lfloor \frac{n}{6} \right\rfloor + 3, & \text{if } n \equiv 5 \pmod 6 \\ 2\left\lfloor \frac{n}{6} \right\rfloor + 2, & \text{otherwise.} \end{cases} \quad (3)$$

There is a similar bound on d_{II}, the minimum distance of a Type II code of length n,

$$d_{\mathrm{II}} \leq 2\left\lfloor \frac{n}{6} \right\rfloor + 2. \tag{4}$$

A code that meets the appropriate bound is called *extremal*. It can be shown that extremal Type II codes must have a unique weight enumerator. Rains and Sloane [2] also used a linear programming bound, and showed that extremal codes do not exist for all lengths. For instance, there is no self-dual $(13, 2^{13}, 6)$ code. If a code has highest possible minimum distance, but is not extremal, it is called *optimal*. An interesting open problem is whether there exists a Type II $(24, 2^{24}, 10)$ code.

A *linear* code, $\mathcal{C}$, over GF(4) which is self-dual with respect to the *Hermitian inner product*, i.e., $\boldsymbol{u} \cdot \overline{\boldsymbol{v}} = 0$ for all $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{C}$, is also a self-dual *additive* code with respect to the *Hermitian trace inner product*. However, most of the self-dual additive codes are not linear. Only Type II codes can be linear, since self-dual linear codes over GF(4) must contain codewords of even weight only. It follows that the set of Hermitian self-dual linear codes over GF(4) is a subset of the set of Type II self-dual additive codes over GF(4).

Example 1. The unique extremal $(6, 2^6, 4)$ code, also known as the *Hexacode*, has a generator matrix

$$\begin{pmatrix} 1 & 0 & 0 & 1 & \omega & \omega \\ \omega & 0 & 0 & \omega & \omega^2 & \omega^2 \\ 0 & 1 & 0 & \omega & 1 & \omega \\ 0 & \omega & 0 & \omega^2 & \omega & \omega^2 \\ 0 & 0 & 1 & \omega & \omega & 1 \\ 0 & 0 & \omega & \omega^2 & \omega^2 & \omega \end{pmatrix}.$$

This code has weight enumerator $W(x, y) = x^6 + 45x^2y^4 + 18y^6$. It is therefore of Type II, and it can be verified that it is also a linear code.

Two self-dual additive codes over GF(4), $\mathcal{C}$ and $\mathcal{C}'$, are *equivalent* if and only if the codewords of $\mathcal{C}$ can be mapped onto the codewords of $\mathcal{C}'$ by a map that preserves self-duality. Such a map must consist of a permutation of coordinates (columns of the generator matrix), followed by multiplication of coordinates by nonzero elements from GF(4), followed by possible conjugation of coordinates. For a code of length n, there is a total of $6^n n!$ such maps. The 6 possible transformations given by scaling and conjugation of a coordinate are equivalent to

the 6 permutations of the elements $\{1, \omega, \omega^2\}$ in the coordinate. A map that maps $\mathcal{C}$ to $\mathcal{C}$ is called an *automorphism* of $\mathcal{C}$. All automorphisms of $\mathcal{C}$ make up an *automorphism group*, denoted $\mathrm{Aut}(\mathcal{C})$. The number of distinct codes equivalent to $\mathcal{C}$ is then given by $\frac{6^n n!}{|\mathrm{Aut}(\mathcal{C})|}$. By summing the sizes of all equivalence classes, we find the total number of distinct codes of length n, denoted T_n. It was shown by Höhn [3] that T_n is also given by the *mass formula*,

$$T_n = \prod_{i=1}^{n} (2^i + 1) = \sum_{j=1}^{t_n} \frac{6^n n!}{|\mathrm{Aut}(\mathcal{C}_j)|}, \tag{5}$$

where the sum is over all equivalence classes. Similarly, the total number of distinct Type II codes of length n is given by

$$T_n^{\mathrm{II}} = \prod_{i=0}^{n-1} (2^i + 1) = \sum_{j=1}^{t_n^{\mathrm{II}}} \frac{6^n n!}{|\mathrm{Aut}(\mathcal{C}_j)|}, \tag{6}$$

where the sum is over the equivalence classes of Type II codes. By assuming that $|\mathrm{Aut}(\mathcal{C}_j)| = 1$ for all j in Eq. (5), we get a lower bound on t_n, the number of inequivalent codes of length n.

$$t_n \geq \left\lceil \frac{\prod_{i=1}^{n} (2^i + 1)}{6^n n!} \right\rceil \tag{7}$$

A similar bound on t_n^{II} can be derived from Eq. (6).

We can use the computational algebra system *Magma* [4] to find the automorphism group of a code. Since, at this time, Magma has no explicit function for calculating the automorphism group of an additive code, we use the following method, described by Calderbank et al. [1]. We map the $(n, 2^k)$ additive code $\mathcal{C}$ over $\mathrm{GF}(4)$ to the $[3n, k]$ binary linear code $\beta(\mathcal{C})$ by applying the map $0 \mapsto 000$, $1 \mapsto 011$, $\omega \mapsto 101$, $\omega^2 \mapsto 110$ to each generator of $\mathcal{C}$. We then use Magma to find $\mathrm{Aut}(\beta(\mathcal{C})) \cap \mathrm{Aut}(\beta(\mathrm{GF}(4)^n))$, which will be isomorphic to $\mathrm{Aut}(\mathcal{C})$.

If we are given t_n inequivalent codes of length n, i.e., one code from each equivalence class, it is relatively easy to calculate the automorphism group size of each code, as described above, and verify that the mass formula defined by Eq. (5) gives the correct value. But to actually find a set of t_n inequivalent codes, or just the value of t_n, is a hard problem. All self-dual additive codes over $\mathrm{GF}(4)$ of length n were first classified, up to equivalence, by Calderbank et al. [1] for

$n \leq 5$ and by Höhn [3] for $n \leq 7$. Höhn also classified all Type II codes of length 8. Using a different terminology, the codes of length n were implicitly classified by Hein, Eisert, and Briegel [5] for $n \leq 7$ and by Glynn et al. [6] for $n \leq 9$. Gaborit et al. [7, 8] have classified all extremal codes of length 8, 9, and 11, and all extremal Type II codes of length 12. Bachoc and Gaborit [9] classified all extremal Type II codes of length 10, and they also showed that there are at least 490 extremal Type II codes of length 14 and gave a partial result on the unicity of the extremal Type II code of length 18. A review of the current status of the classification of various types of self-dual codes was given by Huffman [10].

In this paper, we will give a complete classification of all codes of length up to 12, and all extremal Type II codes of length 14. But first, in Section 2, we introduce *isotropic systems* and show that they correspond to self-dual additive codes over GF(4). It is known that isotropic systems can be represented by graphs. In Section 3 we define *graph codes*. Theorem 6 shows that every code can be represented by a graph. This gives us a much smaller set of objects to work with. In Section 4, we introduce *local complementation*, and Theorem 12 states that two codes are equivalent if and only if the corresponding graphs are related via local complementations and graph isomorphism. This implies that classifying codes up to equivalence is essentially the same as classifying orbits of graphs under local complementation. We describe an algorithm for generating such graph orbits in Section 5. This algorithm was used to classify all codes of length up to 12. We show that Type II codes correspond to a special class of graphs and use this fact to classify all extremal Type II codes of length 14. Finally, we determine that the smallest Type I and Type II codes with trivial automorphism group have length 9 and 12, respectively. In Section 6, we conclude and mention some other results.

2 Isotropic Systems

We define a mapping $\phi : \mathrm{GF}(4) \to \mathrm{GF}(2)^2$ by $\phi(x) = (\mathrm{Tr}(x\omega^2), \mathrm{Tr}(x))$, i.e., $0 \mapsto (0,0)$, $1 \mapsto (1,0)$, $\omega \mapsto (0,1)$ and $\omega^2 \mapsto (1,1)$. The reverse mapping $\phi^{-1} : \mathrm{GF}(2)^2 \to \mathrm{GF}(4)$ is given by $\phi^{-1}(a,b) = a + \omega b$. Let $\boldsymbol{u} \in \mathrm{GF}(2)^{2n}$ be written as $\boldsymbol{u} = (\boldsymbol{a}|\boldsymbol{b}) = (a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n)$. We extend the mapping $\phi : \mathrm{GF}(4)^n \to \mathrm{GF}(2)^{2n}$ by letting $\phi(\boldsymbol{v}) = (\boldsymbol{a}|\boldsymbol{b})$ where $\phi(v_i) = (a_i, b_i)$. Likewise, we define $\phi^{-1} : \mathrm{GF}(2)^{2n} \to \mathrm{GF}(4)^n$ by $\phi^{-1}(\boldsymbol{a}|\boldsymbol{b}) = \boldsymbol{a} + \omega\boldsymbol{b}$. We define the *symplectic inner product*

of $(\boldsymbol{a}|\boldsymbol{b}), (\boldsymbol{a}'|\boldsymbol{b}') \in \mathrm{GF}(2)^{2n}$ as $\langle (\boldsymbol{a}|\boldsymbol{b}), (\boldsymbol{a}'|\boldsymbol{b}') \rangle = \boldsymbol{a} \cdot \boldsymbol{b}' + \boldsymbol{b} \cdot \boldsymbol{a}'$. A subset $\mathcal{I} \subset \mathrm{GF}(2)^{2n}$ is called *totally isotropic* if $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = 0$ for all $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{I}$.

Definition 2. A totally isotropic linear subspace of $\mathrm{GF}(2)^{2n}$ with dimension n defines an *isotropic system* [11]. An isotropic system can therefore be defined by the row space of a full rank $n \times 2n$ binary matrix $(A|B)$, where $AB^{\mathrm{T}} + BA^{\mathrm{T}} = \mathbf{0}$.

Theorem 3. *Every self-dual additive code over* $\mathrm{GF}(4)$ *can be uniquely represented as an isotropic system, and every isotropic system can be uniquely represented as a self-dual additive code over* $\mathrm{GF}(4)$.

Proof. Let $\mathcal{C} \subset \mathrm{GF}(4)^n$ be a self-dual additive code. Map $\mathcal{C}$ to $\mathcal{I} \subset \mathrm{GF}(2)^{2n}$ by mapping each codeword $\boldsymbol{u} \in \mathcal{C}$ to $\phi(\boldsymbol{u}) = (\boldsymbol{a}|\boldsymbol{b}) \in \mathrm{GF}(2)^{2n}$. $\mathcal{I}$ must then be a linear subspace of $\mathrm{GF}(2)^{2n}$ with dimension n. $(\boldsymbol{a}|\boldsymbol{b})$, $(\boldsymbol{a}'|\boldsymbol{b}') \in \mathcal{I}$ are orthogonal with respect to the symplectic inner product if and only if $\phi^{-1}(\boldsymbol{a}|\boldsymbol{b}), \phi^{-1}(\boldsymbol{a}'|\boldsymbol{b}') \in \mathcal{C}$ are orthogonal with respect to the Hermitian trace inner product, because

$$
\begin{aligned}
&\phi^{-1}(\boldsymbol{a}|\boldsymbol{b}) * \phi^{-1}(\boldsymbol{a}'|\boldsymbol{b}') \\
&= \mathrm{Tr}(\phi^{-1}(\boldsymbol{a}|\boldsymbol{b}) \cdot \overline{\phi^{-1}(\boldsymbol{a}'|\boldsymbol{b}')}) \\
&= \mathrm{Tr}((\boldsymbol{a} + \omega\boldsymbol{b}) \cdot (\boldsymbol{a}' + \overline{\omega}\boldsymbol{b}')) \\
&= (\boldsymbol{a} \cdot \boldsymbol{a}') \,\mathrm{Tr}(1) + (\boldsymbol{a} \cdot \boldsymbol{b}') \,\mathrm{Tr}(\overline{\omega}) + (\boldsymbol{b} \cdot \boldsymbol{a}') \,\mathrm{Tr}(\omega) + (\boldsymbol{b} \cdot \boldsymbol{b}') \,\mathrm{Tr}(1) \\
&= \boldsymbol{a} \cdot \boldsymbol{b}' + \boldsymbol{b} \cdot \boldsymbol{a}'.
\end{aligned}
$$

Since $\mathcal{C}$ is self-dual, $\boldsymbol{u} * \boldsymbol{v} = 0$ for all $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{C}$, and $\mathcal{I}$ must therefore be totally isotropic. It follows that $\mathcal{I}$ defines an isotropic system. Likewise, the reverse mapping from an isotropic system to a subset of $\mathrm{GF}(4)^n$ will always give a self-dual additive code over $\mathrm{GF}(4)$. □

Example 4. The row-space of $(A|B)$ defines an isotropic system, while $C = A + \omega B$ is a generator matrix of the $(6, 2^6, 4)$ Hexacode.

$$
(A|B) = \left(\begin{array}{cccccc|cccccc}
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1
\end{array}\right)
$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 1 & \omega & \omega \\ \omega & 0 & 0 & \omega & \omega^2 & \omega^2 \\ 0 & 1 & 0 & \omega & 1 & \omega \\ 0 & \omega & 0 & \omega^2 & \omega & \omega^2 \\ 0 & 0 & 1 & \omega & \omega & 1 \\ 0 & 0 & \omega & \omega^2 & \omega^2 & \omega \end{pmatrix}$$

3 Graph Representation

A *graph* is a pair $G = (V, E)$ where V is a set of *vertices*, and $E \subseteq V \times V$ is a set of *edges*. A graph with n vertices can be represented by an $n \times n$ *adjacency matrix* Γ, where $\gamma_{ij} = 1$ if $\{i, j\} \in E$, and $\gamma_{ij} = 0$ otherwise. We will only consider *simple undirected* graphs whose adjacency matrices are symmetric with all diagonal elements being 0. The *neighbourhood* of $v \in V$, denoted $N_v \subset V$, is the set of vertices connected to v by an edge. The number of vertices adjacent to v, $|N_v|$, is called the *degree* of v. The *induced subgraph* of G on $W \subseteq V$ contains vertices W and all edges from E whose endpoints are both in W. The *complement* of G is found by replacing E with $V \times V - E$, i.e., the edges in E are changed to non-edges, and the non-edges to edges. Two graphs $G = (V, E)$ and $G' = (V, E')$ are *isomorphic* if and only if there exists a permutation π of V such that $\{u, v\} \in E \iff \{\pi(u), \pi(v)\} \in E'$. A *path* is a sequence of vertices, $(v_1, v_2, \ldots, v_i)$, such that $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{i-1}, v_i\} \in E$. A graph is *connected* if there is a path from any vertex to any other vertex in the graph.

Definition 5. A *graph code* is an additive code over GF(4) that has a generator matrix of the form $C = \Gamma + \omega I$, where I is the identity matrix and Γ is the adjacency matrix of a simple undirected graph.

A graph code is always self-dual, since its generator matrix has full rank over GF(2) and $C\overline{C}^{\mathrm{T}}$ only contains entries from GF(2) whose traces must be zero. This construction for self-dual additive codes over GF(4) has also been used by Tonchev [12].

Theorem 6. *Every self-dual additive code over* GF(4) *is equivalent to a graph code.*

Proof. (This proof is due to Van den Nest, Dehaene, and De Moor [13, 14].) We recall that the generator matrix of a self-dual additive code over GF(4) corresponds to an $n \times 2n$ binary matrix $(A|B)$, such that

$C = A + \omega B$. The row-space of $(A|B)$, denoted $\mathcal{I}$, defines an isotropic system. We must prove that $\mathcal{I}$ is also generated by $(\Gamma|I)$, where I is the identity matrix and Γ is the adjacency matrix of a simple undirected graph.

The rows of $(A|B)$ can be replaced by any n independent vectors from $\mathcal{I}$. This basis change can be accomplished by $(A'|B') = M(A|B)$, where M is an $n \times n$ invertible binary matrix. If B is invertible, the solution is simple, since $B^{-1}(A|B) = (\Gamma|I)$. Note that Γ will always be a symmetric matrix, since $\Gamma I^{\mathrm{T}} + I\Gamma^{\mathrm{T}} = 0$. If the i-th diagonal element of Γ is 1, it can be set to 0 by conjugating column i of $\Gamma + \omega I$.

In the case where B has rank $k < n$, we can perform a basis change to get

$$(A'|B') = \left(\begin{array}{c|c} A_1 & B_1 \\ A_2 & \mathbf{0} \end{array}\right),$$

where B_1 is a $k \times n$ matrix with full rank, and A_1 also has size $k \times n$. Since the row-space of $(A'|B')$ is totally isotropic, and B' contains an all-zero row, it must be true that $A_2 B_1^{\mathrm{T}} = \mathbf{0}$. A_2 must have full rank, and the row space of B_1 must be the orthogonal complement of the row space of A_2.

We assume that $B_1 = (B_{11}|B_{12})$ where B_{11} is a $k \times k$ invertible matrix. We also write $A_2 = (A_{21}|A_{22})$ where A_{22} has size $(n-k) \times (n-k)$. Assume that there exists an $\boldsymbol{x} \in \mathrm{GF}(2)^{n-k}$ such that $A_{22}\boldsymbol{x}^{\mathrm{T}} = 0$. Then the vector $\boldsymbol{v} = (0, \ldots, 0, \boldsymbol{x})$ of length n satisfies $A_2\boldsymbol{v}^{\mathrm{T}} = 0$. Since the row space of B_1 is the orthogonal complement of the row space of A_2, we can write $\boldsymbol{v} = \boldsymbol{y}B_1$ for some $\boldsymbol{y} \in \mathrm{GF}(2)^k$. We see that $\boldsymbol{y}B_{11} = 0$, and since B_{11} has full rank, it must therefore be true that $\boldsymbol{y} = 0$. This means that $\boldsymbol{x} = 0$, which proves that A_{22} is an invertible matrix.

Interchanging column i of A' and column i of B' corresponds to multiplication by ω^2 followed by conjugation of the i-th column of $A' + \omega B'$. We can therefore swap the i-th columns of A' and B' for $k < i \leq n$ to get $(A''|B'')$. Since B_{11} and A_{22} are invertible, B'' must also be an invertible matrix. We then find $B''^{-1}(A''|B'') = (\Gamma|I)$, and set all diagonal elements of Γ to 0. □

Example 7. Let $C = A + \omega B$ be the generator matrix of the $(6, 2^6, 4)$ Hexacode given in Example 4. By the method described in the proof of

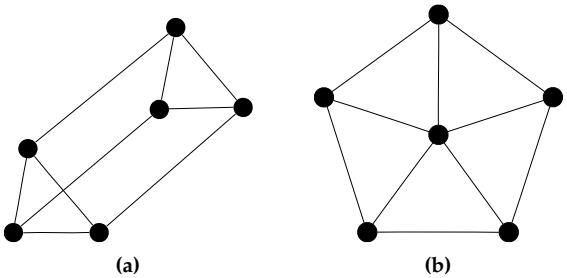


Fig. 1: *Two Graph Representations of the Hexacode*

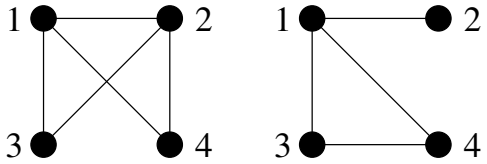
Theorem 6, we find $C' = \Gamma + \omega I$, which generates an equivalent graph code. Γ is the adjacency matrix of the graph shown in Fig. 1b.

$$(A|B) = \left(\begin{array}{cccccc|cccccc} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right)$$

$$(\Gamma|I) = \left(\begin{array}{cccccc|cccccc} 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

$$C' = \begin{pmatrix} \omega & 0 & 1 & 0 & 1 & 1 \\ 0 & \omega & 1 & 1 & 0 & 1 \\ 1 & 1 & \omega & 0 & 0 & 1 \\ 0 & 1 & 0 & \omega & 1 & 1 \\ 1 & 0 & 0 & 1 & \omega & 1 \\ 1 & 1 & 1 & 1 & 1 & \omega \end{pmatrix}$$

Theorem 6 was first proved by Bouchet [15] in the context of isotropic systems, and later by Schlingemann [16] in terms of *quantum stabilizer states*. Proofs of Theorem 6 have also been given by Schlingemann and Werner [17], by Grassl, Klappenecker, and Rötteler [18], by Glynn et al. [6, 19], and by Van den Nest et al. [13, 14].



(a) *The Graph G* **(b)** *The LC Image G ∗ 1*

Fig. 2: *Example of Local Complementation*

Swapping vertex i and vertex j of a graph with adjacency matrix Γ can be accomplished by exchanging column i and column j of Γ and then exchanging row i and row j of Γ. We call the resulting matrix Γ'. Exactly the same column and row operations map $\Gamma + \omega I$ to $\Gamma' + \omega I$. These matrices generate equivalent codes. It follows that two codes are equivalent if their corresponding graphs are isomorphic.

We have seen that every graph represents a self-dual additive code over GF(4), and that every self-dual additive code over GF(4) can be represented by a graph. It follows that we can, without loss of generality, restrict our study to codes with generator matrices of the form $\Gamma + \omega I$, where Γ are adjacency matrices of unlabeled simple undirected graphs.

4 Local Complementation

Definition 8. Given a graph $G = (V, E)$ and a vertex $v \in V$, let $N_v \subset V$ be the neighbourhood of v. *Local complementation* (LC) on v transforms G into $G * v$. To obtain $G * v$, we replace the induced subgraph of G on N_v by its complement. It is easy to verify that $G * v * v = G$.

Example 9. We will perform local complementation on vertex 1 of the graph G, shown in Fig. 2a. We see that the neighbourhood of 1 is $N_1 = \{2, 3, 4\}$ and that the induced subgraph on the neighbourhood has edges $\{2, 3\}$ and $\{2, 4\}$. The complement of this subgraph contains the single edge $\{3, 4\}$. The resulting LC image, $G * 1$, is seen in Fig. 2b.

Example 10. Consider the graph shown in Fig. 1a, whose corresponding graph code is the Hexacode. An LC operation on any vertex of this graph produces the graph shown in Fig. 1b. An LC operation on the vertex in the centre of the graph shown in Fig. 1b gives the same graph, up to isomorphism. LC operations on any of the other five vertices produces the graph shown in Fig. 1a.

Theorem 11. *Let Γ be the adjacency matrix of the graph $G = (V, E)$, and Γ' be the adjacency matrix of $G * v$, for any $v \in V$. The codes generated by $C = \Gamma + \omega I$ and $C' = \Gamma' + \omega I$ are equivalent.*

Proof. We show that C can be transformed into C' by using only operations that map a code to an equivalent code. Each row and each column of C correspond to a vertex in V. Let N_v denote the neighbourhood of v. For all $i \in N_v$, add row v of C to row i of C. Multiply column v of C by ω and then conjugate the same column. Finally, conjugate column i of C, for all $i \in N_v$. The resulting matrix is C'. □

Theorem 12. *Two self-dual additive codes over* GF(4)*, $\mathcal{C}$ and $\mathcal{C}'$, with graph representations G and G', are equivalent if and only if there is a finite sequence of not necessarily distinct vertices $(v_1, v_2, \ldots, v_i)$, such that $G * v_1 * v_2 * \cdots * v_i$ is isomorphic to G'.*

Sketch of proof. Let Γ be the adjacency matrix of G, and let $\mathcal{C}_G$ be the code generated by $\Gamma + \omega I$. Likewise, let Γ' be the adjacency matrix of G', and let $\mathcal{C}'_G$ be the code generated by $\Gamma' + \omega I$. If the codewords of $\mathcal{C}$ are mapped onto the codewords of $\mathcal{C}'$ by one of the $6^n n!$ combinations of coordinate permutations, coordinate scalings, and coordinate conjugations, then there must also be a transformation from this set that maps the codewords of $\mathcal{C}_G$ onto the codewords of $\mathcal{C}'_G$. Consequently, we only need to consider those transformations that map a graph code to another graph code. The codes obtained by the $n!$ possible permutations of coordinates correspond to graph isomorphisms.

Let $C = \Gamma + \omega I$ be transformed into $C' = A + \omega B$ by coordinate scalings and conjugations. Then C' is a graph code if and only if B is invertible and all diagonal elements of $B^{-1}A$ are zero. It is easy to verify that conjugation of column i of C' has no effect on B, but flips the value of the i-th diagonal element of $B^{-1}A$. Given a combination of column scalings on C such that the resulting B is invertible, there must therefore be a unique combination of column conjugations on C such that the resulting $B^{-1}A$ has zero diagonal. We must therefore show that any combination of column scalings on C that give an invertible B can be performed by a sequence of LC operations on G.

Multiplying column i of C by ω^2 replaces column i of I with column i of Γ. Multiplying column i of C by ω adds column i of Γ to column i of I. It is then possible to show which of the 3^n possible scalings do not give an invertible B. A vertex v of G corresponds to a column of Γ. The neighbourhood of v, N_v, corresponds to a set of columns of Γ. We know

from Theorem 11 that an LC operation on vertex i of G corresponds to a scaling of column i of C by ω followed by conjugation of column i and all columns in N_i. Observe that conjugating a coordinate followed by a scaling by ω is equivalent to scaling by ω^2 followed by conjugation. Note in particular that the local complementations $G * i * j * i$, where i and j are adjacent vertices, are equivalent to scaling both column i and column j of C by ω^2. It can be verified that any combination of column scalings that map a graph code to a graph code can be implemented as a sequence of LC operations. The exact algorithm for finding this sequence of LC operations is quite involved, and we refer to the proof by Van den Nest et al. [13, 14] for the details. □

Bouchet [15] first proved Theorem 12 in terms of isotropic systems. The same result was discovered by Van den Nest et al. [13, 14] in terms of quantum stabilizer states, and by Glynn et al. [6, 19] using finite geometry.

5 Classification

Definition 13. The *LC orbit* of a graph G is the set of all unlabeled graphs that can be obtained by performing any sequence of LC operations on G.

It follows from Theorem 12 that two self-dual additive codes over $\mathrm{GF}(4)$ are equivalent if and only if their graph representations are in the same LC orbit. As an example, the two graphs shown in Fig. 1a and Fig. 1b make up a complete LC orbit, and are thus the only possible graph representations of the Hexacode. The LC orbit of a graph can easily be generated by a recursive algorithm. We have used the program *nauty* [20] to check for graph isomorphism.

Let $\boldsymbol{G}_n$ be the set containing all unlabeled simple undirected connected graphs on n vertices. Connected graphs correspond to *indecomposable* codes. A code is decomposable if it can be written as the *direct sum* of two smaller codes. For example, let $\mathcal{C}$ be an $(n, 2^n, d)$ code and $\mathcal{C}'$ an $(n', 2^{n'}, d')$ code. The direct sum, $\mathcal{C} \oplus \mathcal{C}' = \{u||v \mid u \in \mathcal{C}, v \in \mathcal{C}'\}$, where $||$ means concatenation, is an $(n+n', 2^{n+n'}, \min\{d, d'\})$ code. It follows that all decomposable codes of length n can be classified easily once all indecomposable codes of length less than n are known.

The set of all distinct LC orbits of connected graphs on n vertices is a partitioning of $\boldsymbol{G}_n$ into i_n disjoint sets. i_n is also the number of

indecomposable self-dual additive codes over GF(4) of length n, up to equivalence. Let $\boldsymbol{L}_n$ be a set containing one representative from each LC orbit of connected graphs on n vertices. We have devised several algorithms [21] for finding such sets of representatives. The simplest approach is to start with the set $\boldsymbol{G}_n$ and generate LC orbits of its members until we have a partitioning of $\boldsymbol{G}_n$. The following more efficient technique was described by Glynn et al. [6]. Let the $2^n - 1$ *extensions* of a graph on n vertices be formed by adding a new vertex and joining it to all possible combinations of at least one of the old vertices. The set $\boldsymbol{E}_n$, containing $i_{n-1}(2^{n-1} - 1)$ graphs, is formed by making all possible extensions of all graphs in $\boldsymbol{L}_{n-1}$.

Theorem 14. *$\boldsymbol{L}_n \subset \boldsymbol{E}_n$, i.e., the set $\boldsymbol{E}_n$ will contain at least one representative from each LC orbit of connected graphs on n vertices.*

Proof. Let $G = (V, E) \in \boldsymbol{G}_n$, and choose any subset $W \subset V$ of $n - 1$ vertices. By doing LC operations on vertices in W, we can transform the induced subgraph of G on W into one of the graphs in $\boldsymbol{L}_{n-1}$ that were extended when $\boldsymbol{E}_n$ was constructed. It follows that for all $G \in \boldsymbol{G}_n$, some graph in the LC orbit of G must be part of $\boldsymbol{E}_n$. □

The set $\boldsymbol{E}_n$ will be much smaller than $\boldsymbol{G}_n$, so it will be more efficient to search for a set of LC orbit representatives within $\boldsymbol{E}_n$. It is also desirable to partition the set $\boldsymbol{E}_n$ such that graphs from two different partitions are guaranteed to belong to different LC orbits. We can then consider each partition independently, which reduces the amount of memory required and allow for parallel processing. To do this, we must have some property that is invariant over the LC orbit and that can be calculated quickly.

The special form of the generator matrix of a graph code makes it easier to find the number of codewords of weight $i < n$. If $\mathcal{C}$ is generated by $C = \Gamma + \omega I$, then any codeword formed by adding i rows of C must have weight at least i. This means that we can find the *partial weight distribution* of $\mathcal{C}$, $(A_0, A_1, \ldots, A_j)$, for some $j < n$, by only considering codewords formed by adding j or fewer rows of C. We calculate the partial weight distribution, for a suitable choice of j, of all codes corresponding to graphs in $\boldsymbol{E}_n$. Codes with different partial weight distribution can never be equivalent, so we partition $\boldsymbol{E}_n$ such that graphs corresponding to codes with the same partial weight distribution are always in the same partition.

Using the described techniques, and a parallel cluster computer, we were able to classify all self-dual additive codes over GF(4) of length

up to 12. The results have been verified by checking that the sizes of all LC orbits add up to the number of graphs in $\boldsymbol{G}_n$. The sizes of the automorphism groups of all codes have also been calculated, and it has been verified that that the mass formulas defined by Eq. (5) and Eq. (6) give the correct values. Table 1 gives the values of i_n, the number of distinct LC orbits of connected graphs on n vertices, which is also the number of inequivalent indecomposable codes of length n. The table also gives the values of i_n^{II}, the number of indecomposable Type II codes. The total number of inequivalent codes of length n, t_n, and the total number of Type II codes of length n, t_n^{II}, are shown in Table 2. The numbers t_n are easily derived from the numbers i_n by using the *Euler transform* [22],

$$\begin{aligned} c_n &= \sum_{d|n} d i_d \\ t_1 &= c_1 \\ t_n &= \frac{1}{n}\left(c_n + \sum_{k=1}^{n-1} c_k t_{n-k}\right). \end{aligned}$$

The numbers t_n^{II} are similarly derived from i_n^{II}. The values of i_n and t_n can be found as sequences A090899 and A094927 in *The On-Line Encyclopedia of Integer Sequences* [23]. Table 3 and Table 4 list by minimum distance the numbers of indecomposable codes and the total numbers of codes.[1] Table 5 and Table 6 similarly list the numbers of Type II codes by minimum distance. The numbers of Type I codes can be obtained by subtracting the numbers of Type II codes from the total numbers. The number of distinct weight enumerators of all codes of length n and minimum distance d can be found in Table 7. There are obviously too many codes to give a complete list here, but a database containing one representative from each equivalence class, with information about weight enumerators, automorphism groups, etc., is available on-line at `http://www.ii.uib.no/~larsed/vncorbits/`.

Our results give a complete classification of the extremal Type I $(10, 2^{10}, 4)$ and $(12, 2^{12}, 5)$ codes. These classifications were previously unknown. The 101 extremal Type I $(10, 2^{10}, 4)$ codes have 6 distinct

[1] Note that some authors [7, 10] give 3 as the total number of self-dual $(7, 2^7, 3)$ codes. The correct number is 4.

Table 1: *Number of Indecomposable (i_n) and Indecomposable Type II (i_n^{II}) Codes of Length n*

n	1	2	3	4	5	6	7	8	9	10	11	12
i_n	1	1	1	2	4	11	26	101	440	3132	40 457	1 274 068
i_n^{II}		1		1		4		14		103		2926

Table 2: *Total Number (t_n) and Number of Type II (t_n^{II}) Codes of Length n*

n	1	2	3	4	5	6	7	8	9	10	11	12
t_n	1	2	3	6	11	26	59	182	675	3990	45 144	1 323 363
t_n^{II}		1		2		6		21		128		3079

Table 3: *Number of Indecomposable Codes of Length n and Minimum Distance d*

$d\backslash n$	2	3	4	5	6	7	8	9	10	11	12
2	1	1	2	3	9	22	85	363	2436	26 750	611 036
3				1	1	4	11	69	576	11 200	467 513
4					1		5	8	120	2506	195 455
5										1	63
6											1
All	1	1	2	4	11	26	101	440	3132	40 457	1 274 068

Table 4: *Total Number of Codes of Length n and Minimum Distance d*

$d\backslash n$	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	2	3	6	11	26	59	182	675	3990	45 144
2		1	1	3	4	13	29	107	416	2618	27 445	615 180
3					1	1	4	11	69	577	11 202	467 519
4						1		5	8	120	2506	195 456
5											1	63
6												1
All	1	2	3	6	11	26	59	182	675	3990	45 144	1 323 363

Table 5: *Number of Indecomposable Type II Codes of Length n and Minimum Distance d*

$d\backslash n$	2	4	6	8	10	12	14
2	1	1	3	11	84	2133	?
4			1	3	19	792	?
6						1	1020
Total	1	1	4	14	103	2926	?

Table 6: *Total Number of Type II Codes of Length n and Minimum Distance d*

$d\backslash n$	2	4	6	8	10	12	14
2	1	2	5	18	109	2285	?
4			1	3	19	793	?
6						1	1020
Total	1	2	6	21	128	3079	$\geq$ 1 727 942

Table 7: *Number of Distinct Weight Enumerators of All Codes of Length n and Minimum Distance d*

$d\backslash n$	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	2	3	5	10	23	46	116	320	909	3312
2		1	1	2	4	11	21	64	187	549	2249	11 419
3					1	1	2	4	15	33	125	625
4						1		2	2	7	28	178
5											1	2
6												1
All	1	2	3	5	10	23	46	116	320	909	3312	15 537

weight enumerators,

$$W_{10,1}(x,y) = x^{10} + 10x^6y^4 + 72x^5y^5 + 160x^4y^6 + 240x^3y^7 + 285x^2y^8 + 200xy^9 + 56y^{10},$$
$$W_{10,2}(x,y) = x^{10} + 14x^6y^4 + 64x^5y^5 + 156x^4y^6 + 256x^3y^7 + 281x^2y^8 + 192xy^9 + 60y^{10},$$
$$W_{10,3}(x,y) = x^{10} + 18x^6y^4 + 56x^5y^5 + 152x^4y^6 + 272x^3y^7 + 277x^2y^8 + 184xy^9 + 64y^{10},$$
$$W_{10,4}(x,y) = x^{10} + 22x^6y^4 + 48x^5y^5 + 148x^4y^6 + 288x^3y^7 + 273x^2y^8 + 176xy^9 + 68y^{10},$$
$$W_{10,5}(x,y) = x^{10} + 26x^6y^4 + 40x^5y^5 + 144x^4y^6 + 304x^3y^7 + 269x^2y^8 + 168xy^9 + 72y^{10},$$
$$W_{10,6}(x,y) = x^{10} + 30x^6y^4 + 32x^5y^5 + 140x^4y^6 + 320x^3y^7 + 265x^2y^8 + 160xy^9 + 76y^{10}.$$

Table 8 lists the number of such codes by weight enumerator and automorphism group size. The 63 extremal Type I $(12, 2^{12}, 5)$ codes have 2 distinct weight enumerators,

$$\begin{aligned} W_{12,1}(x,y) = {} & x^{12} + 40x^7y^5 + 212x^6y^6 + 424x^5y^7 + 725x^4y^8 + \\ & 1080x^3y^9 + 980x^2y^{10} + 504xy^{11} + 130y^{12}, \\ W_{12,2}(x,y) = {} & x^{12} + 48x^7y^5 + 188x^6y^6 + 432x^5y^7 + 765x^4y^8 + \\ & 1040x^3y^9 + 972x^2y^{10} + 528xy^{11} + 122y^{12}. \end{aligned}$$

Table 9 lists the number of such codes by weight enumerator and automorphism group size.

By observing that graphs corresponding to Type II codes have a special property, we are able to extend our classification to all the 1020 extremal Type II $(14, 2^{14}, 6)$ codes. It was previously shown by Bachoc and Gaborit [9] that there are at least 490 such codes.

Theorem 15. *Let Γ be the adjacency matrix of the graph G. The code $\mathcal{C}$ generated by $C = \Gamma + \omega I$ is of Type II if and only if G is* anti-Eulerian*, i.e., if all its vertices have odd degree.*

Proof. If $\mathcal{C}$ is of Type II, then every row of C must have even weight. It follows that every row of Γ must have odd weight, and therefore correspond to an anti-Eulerian graph. Conversely, if all rows of C have even weight, $\mathcal{C}$ must be of Type II, since the codeword formed by adding any subset of these rows must also have even weight. This follows from the fact that for any two codewords of a self-dual code, there must be an even number of coordinates where the codewords have different non-zero values. □

An anti-Eulerian graph is the complement of an *Eulerian graph*, i.e., a graph where all vertices have even degree. It is easy to show that all anti-Eulerian graphs must have an even number of vertices, and

Table 8: *Number of Extremal Type I* $(10, 2^{10}, 4)$ *Codes with Weight Enumerator* w *and Automorphism Group of Size* a

$a \backslash w$	$W_{10,1}$	$W_{10,2}$	$W_{10,3}$	$W_{10,4}$	$W_{10,5}$	$W_{10,6}$	All
1		3					3
2	2	9	7	2			20
4	5	9	7	1			22
6	1			1			2
8	1	4	3		1		9
12		1					1
16	1	1	6	5	3		16
32	2	2	2	1	2		9
40	1						1
48	1			3			4
64			2				2
128		2					2
192		1	2		1		4
256						2	2
320	1					1	2
384						1	1
3840						1	1
All	15	32	29	13	7	5	101

Table 9: *Number of Extremal Type I* $(12, 2^{12}, 5)$ *Codes with Weight Enumerator* w *and Automorphism Group of Size* a

$a \backslash w$	$W_{12,1}$	$W_{12,2}$	All
1		25	25
2		23	23
3		1	1
4	3	4	7
6	1	3	4
8		2	2
24		1	1
All	4	59	63

it follows that all Type II codes must have even length. To classify Type II codes of length 14, we proceed as follows. We take the set $\boldsymbol{L}_{12}$ containing 1 274 068 LC orbit representatives of graphs on 12 vertices. All these graphs are then extended, but in a slightly different way than earlier. To each graph we add one vertex and join it to all possible combinations of at least one of the old vertices. To each obtained graph we then add a second vertex and join it to those of the 13 other vertices that have even degree. (If the result is not a connected anti-Eulerian graph, it is rejected.) By an argument similar to Theorem 14, it can be shown that all graphs corresponding to Type II codes of length 14 must be part of this extended set. Classifying all Type II codes of length 14 turned out to be infeasible with our computational resources. Even when using partitioning by partial weight distribution, the largest partitions were too large to be processed. However, we were able to generate the LC orbits of all graphs corresponding to $(14, 2^{14}, 6)$ codes. Extremal Type II codes have a unique weight enumerator, and the weight enumerator of a $(14, 2^{14}, 6)$ code must be

$$W_{14}(x, y) = x^{14} + 273x^8y^6 + 2457x^6y^8 + 7098x^4y^{10} + 6006x^2y^{12} + 549y^{14}.$$

Table 10 lists the number of codes by automorphism group size. Note that codes with 21, 168, and 2184 automorphisms were previously unknown. Generator matrices of the codes are available on-line at `http://www.ii.uib.no/~larsed/vncorbits/`.

As mentioned before, the set of self-dual *linear* codes over GF(4) is a subset of the self-dual additive codes of Type II. Note that conjugation of single coordinates does not preserve the linearity of a code. It was shown by Van den Nest [13] that the code $\mathcal{C}$ generated by a matrix of the form $\Gamma + \omega I$ can not be linear. However, if there is a linear code equivalent to $\mathcal{C}$, it can be found by conjugating some coordinates. Conjugating coordinates of $\mathcal{C}$ is equivalent to setting some diagonal elements of Γ to 1. Let A be a binary diagonal matrix such that $\Gamma + A + \omega I$ generates a linear code. Van den Nest [13] proved that $\mathcal{C}$ is equivalent to a linear code if and only if there exists such a matrix A that satisfies $\Gamma^2 + A\Gamma + \Gamma A + \Gamma + I = 0$. A similar result was found by Glynn et al. [6]. Using this method, it is easy to check whether the LC orbit of a given graph corresponds to a linear code. However, self-dual linear codes over GF(4) have already been classified up to length 16, and we have not found a way to extend this result using the graph approach.

Table 10: *Number of* $(14, 2^{14}, 6)$ *Codes with Automorphism Group of Size a*

a	
1	625
2	258
3	27
4	38
6	27
8	13
12	7
18	1
21	1
24	16
28	1
36	1
48	1
84	1
168	1
2184	1
6552	1
All	1020

We remark that if $\mathcal{C}$ is a self-dual additive code over GF(4) with generator matrix $\Gamma + \omega I$, it can be shown that the additive code over $\mathbb{Z}_4$ generated by $2\Gamma + I$ has the same weight distribution as $\mathcal{C}$. It has also been shown [2] that self-dual additive codes over GF(4) can be mapped to *isodual* binary linear codes, i.e., codes that are equivalent to their duals, by the mapping $0 \mapsto 00$, $1 \mapsto 11$, $\omega \mapsto 01$ and $\omega^2 \mapsto 10$. A code over $\mathbb{Z}_4$ and a binary code obtained from the same self-dual additive code over GF(4) by these two methods are related by the well-known *Gray map*. There are also several mappings from self-dual additive codes over GF(4) to self-dual and self-orthogonal binary linear codes [3, 7, 24].

An interesting problem, posed by Höhn [3], is to find the smallest code with trivial automorphism group, i.e., automorphism group of size 1. We find that there is no such code of length less than 9, but there is a single code of length 9 with trivial automorphism group. This code has generator matrix

$$\begin{pmatrix}
\omega & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & \omega & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & \omega & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & \omega & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & \omega & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & \omega & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & \omega & 0 & 1 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 & \omega & 0 \\
1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & \omega
\end{pmatrix}.$$

The smallest Type II codes with trivial automorphism group have length 12. One such code is generated by

$$\begin{pmatrix}
\omega & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & \omega & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & \omega & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & \omega & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & \omega & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & \omega & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 & \omega & 0 & 0 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & \omega & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & \omega & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \omega & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & \omega & 0 \\
1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & \omega
\end{pmatrix}.$$

Table 11: *Number of Type I (Type II) Codes of Length n and Minimum Distance d with Trivial Automorphism Group*

$d\backslash n$	≤ 8	9	10	11	12	14
3		1 (0)	113 (0)	6247 (0)	392 649 (0)	? (0)
4			3 (0)	1180 (0)	163 982 (102)	? (?)
5					25 (0)	? (0)
6						? (625)
All	0 (0)	1 (0)	116 (0)	7427 (0)	556 656 (102)	? (?)

Table 11 lists the numbers of Type I and Type II codes with trivial automorphism group by length and minimum distance. Note that for length 12, almost half the codes have trivial automorphism group. For high lengths, one can expect almost all codes to have trivial automorphism group [3]. This implies that the bound on t_n given by Eq. (7) is tighter for higher n. Observe that in Table 11, no code of minimum distance less than 3 is listed. It is easy to show that all codes with minimum distance 1 or 2 must have nontrivial automorphisms.

6 Conclusions

By using graph representation and equivalence via local complementation, we have classified all additive codes over GF(4) of length up to 12 that are self-dual with respect to the Hermitian trace inner product. It follows from the bound given by Eq. (7) that there are at least 72 573 549 codes of length 13. It is not feasible to classify all codes of length 13 using our method and the computational resources available to us. We were however able to classify the 1020 extremal Type II $(14, 2^{14}, 6)$ codes. This was done by exploiting the fact that Type II codes correspond to anti-Eulerian graphs. Finally, we showed that the smallest Type I and Type II codes with trivial automorphism group have length 9 and 12, respectively.

The graph representation of a self-dual additive code over GF(4) can also give information about the properties of the code. Tonchev [12] showed that *strongly regular* graphs give rise to interesting codes. In particular, codes represented by the strongly regular *Paley graphs* are well-known *quadratic residue codes*. We have shown that many extremal and optimal codes can be represented by *nested regular graphs* [21, 25].

Glynn et al. [6] showed that the minimum distance of a code is equal to one plus the minimum *vertex degree* over all graphs in the corresponding LC orbit. We have shown that the LC orbit corresponding to a code with high minimum distance only contains graphs with both small *independent sets* and small *cliques* [21, 25].

ACKNOWLEDGEMENTS We would like to thank Philippe Gaborit for his helpful comments. Also thanks to the Bergen Center for Computational Science, whose cluster computer made the results in this paper possible. This research was supported by the Research Council of Norway.

REFERENCES

[1] CALDERBANK, A. R., RAINS, E. M., SHOR, P. M., SLOANE, N. J. A.: Quantum error correction via codes over GF(4). *IEEE Trans. Inform. Theory* 44(4), 1369–1387, 1998. arXiv:quant-ph/9608006.

[2] RAINS, E. M., SLOANE, N. J. A.: Self-dual codes. In *Handbook of Coding Theory*, pp. 177–294. North-Holland, Amsterdam, 1998. arXiv:math.CO/0208001.

[3] HÖHN, G.: Self-dual codes over the Kleinian four group. *Math. Ann.* 327(2), 227–255, 2003. arXiv:math.CO/0005266.

[4] CANNON, J., BOSMA, W.: *Handbook of Magma functions, Version 2.11*, May 2004. `http://magma.maths.usyd.edu.au/`.

[5] HEIN, M., EISERT, J., BRIEGEL, H. J.: Multi-party entanglement in graph states. *Phys. Rev. A* 69(6), 2004. arXiv:quant-ph/0307130.

[6] GLYNN, D. G., GULLIVER, T. A., MAKS, J. G., GUPTA, M. K.: The geometry of additive quantum codes, 2004. Submitted to Springer.

[7] GABORIT, P., HUFFMAN, W. C., KIM, J.-L., PLESS, V.: On additive GF(4) codes. In *Codes and Association Schemes, DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, vol. 56, pp. 135–149. Amer. Math. Soc., Providence, RI, 2001.

[8] GABORIT, P., HUFFMAN, W. C., KIM, J.-L., PLESS, V.: On the classification of extremal additive codes over GF(4). In *Proc. of the 37th Allerton Conference on Communication, Control, and Computing*, pp. 535–544. Univ. Illinois at Urbana-Champaign, 1999.

[9] BACHOC, C., GABORIT, P.: On extremal additive $\mathbb{F}_4$ codes of length 10 to 18. *J. Théor. Nombres Bordeaux* 12(2), 255–271, 2000.

[10] Huffman, W. C.: On the classification and enumeration of self-dual codes. *Finite Fields Appl.* 11(3), 451–490, 2005.

[11] Bouchet, A.: Isotropic systems. *European J. Combin.* 8(3), 231–244, 1987.

[12] Tonchev, V. D.: Error-correcting codes from graphs. *Discrete Math.* 257(2–3), 549–557, 2002.

[13] Van den Nest, M.: *Local Equivalence of Stabilizer States and Codes*. Ph.D. thesis, K. U. Leuven, Belgium, May 2005.

[14] Van den Nest, M., Dehaene, J., De Moor, B.: Graphical description of the action of local Clifford transformations on graph states. *Phys. Rev. A* 69(2), 2004. arXiv:quant-ph/0308151.

[15] Bouchet, A.: Graphic presentations of isotropic systems. *J. Combin. Theory Ser. B* 45(1), 58–76, 1988.

[16] Schlingemann, D.: Stabilizer codes can be realized as graph codes. *Quantum Inf. Comput.* 2(4), 307–323, 2002. arXiv:quant-ph/0111080.

[17] Schlingemann, D., Werner, R. F.: Quantum error-correcting codes associated with graphs. *Phys. Rev. A* 65(1), 2002. arXiv:quant-ph/0012111.

[18] Grassl, M., Klappenecker, A., Rötteler, M.: Graphs, quadratic forms, and quantum codes. In *Proc. IEEE Int. Symp. Inform. Theory*, p. 45. 2002. arXiv:quant-ph/0703112.

[19] Glynn, D. G.: On self-dual quantum codes and graphs, 2002. Submitted to Electron. J. Combin.

[20] McKay, B. D.: *nauty User's Guide, Version 2.2*, 2003. `http://cs.anu.edu.au/~bdm/nauty/`.

[21] Danielsen, L. E.: *On Self-Dual Quantum Codes, Graphs, and Boolean Functions*. Master's thesis, Dept. Informat., Univ. Bergen, Norway, Mar. 2005. arXiv:quant-ph/0503236.

[22] Sloane, N. J. A., Plouffe, S.: *The Encyclopedia of Integer Sequences*. Academic Press, San Diego, CA, 1995.

[23] Sloane, N. J. A.: The On-Line Encyclopedia of Integer Sequences. `http://www.research.att.com/~njas/sequences/`.

[24] Kim, J.-L., Mellinger, K. E., Pless, V.: Projections of binary linear codes onto larger fields. *SIAM J. Discrete Math.* 16(4), 591–603, 2003.

[25] Danielsen, L. E., Parker, M. G.: Spectral orbits and peak-to-average power ratio of Boolean functions with respect to the $\{I, H, N\}^n$ transform. In *Sequences and Their Applications – SETA 2004, Lecture Notes in Comput. Sci.*, vol. 3486, pp. 373–388. Springer, Berlin, 2005. arXiv:cs.IT/0504102.

PAPER II

GRAPH-BASED CLASSIFICATION OF SELF-DUAL ADDITIVE CODES OVER FINITE FIELDS

Lars Eirik Danielsen

GRAPH-BASED CLASSIFICATION OF SELF-DUAL ADDITIVE CODES OVER FINITE FIELDS

Lars Eirik Danielsen*

Quantum stabilizer states over GF(m) can be represented as self-dual additive codes over GF(m^2). These codes can be represented as weighted graphs, and orbits of graphs under the generalized local complementation operation correspond to equivalence classes of codes. We have previously used this fact to classify self-dual additive codes over GF(4). In this paper we classify self-dual additive codes over GF(9), GF(16), and GF(25). Assuming that the classical MDS conjecture holds, we are able to classify all self-dual additive MDS codes over GF(9) by using an extension technique. Circulant graph codes are introduced, and a computer search reveals that this set contains many strong codes. We show that some of these codes have highly regular graph representations.

1 INTRODUCTION

It is well-known that *self-orthogonal additive codes* over GF(4) can be used to represent a class of *quantum error-correcting codes* known as *binary stabilizer codes* [1]. Although the binary stabilizer codes have been studied most, several authors have considered nonbinary stabilizer codes over finite fields [2–7], cyclic groups [8], and Abelian groups in general [9]. We will focus mainly on codes over finite fields, and

*Department of Informatics, University of Bergen, PB 7803, N-5020 Bergen, Norway.

exploit the fact that a stabilizer code over GF(m) corresponds to a self-orthogonal *additive code* over GF(m^2). Quantum codes of dimension zero are known as *stabilizer states*, which are entangled quantum states with several possible applications. Stabilizer states correspond to *self-dual* additive codes. It is known that such codes can be represented as graphs [6, 9]. It is also known that two self-dual additive codes over GF(4) are equivalent if and only if their corresponding graphs are equivalent, up to isomorphism, with respect to a sequence of *local complementations* [10–13]. We have previously used this fact to devise a graph-based algorithm with which we classified all self-dual additive codes over GF(4) of length up to 12 [14]. Recently, the representation of equivalence classes as graph orbits was generalized to self-dual additive codes over any finite field [15]. In this paper we use graph-based algorithms to classify all self-dual additive codes over GF(9), GF(16), and GF(25) up to lengths 8, 6, and 6, respectively. We also give upper bounds on the number of codes, derived from *mass formulas*. By using a graph extension technique we find that there are only 3 self-dual additive MDS codes over GF(9), assuming that the classical MDS conjecture holds. We prove that the minimum distance of a self-dual additive code is related to the minimum vertex degree in the associated graph orbit. Finally, we perform a search of *circulant graph codes*, a subclass of the self-dual additive codes, which is shown to contain many codes with high minimum distance. The highly regular graph structures of some of these codes are described.

2 Stabilizer States

Data in a classical computer is typically stored in bits that have either the values 0 or 1. Similarly, we can envisage a quantum computer where data is stored in quantum bits, also known as *qubits*, i.e., two-level quantum systems. One qubit can then be described by a vector $|x\rangle = \binom{\alpha}{\beta} \in \mathbb{C}^2$, where $|\alpha|^2$ is the probability of observing the value 0 when we measure the qubit, and $|\beta|^2$ is the probability of observing the value 1. More generally, data could be stored in m-level *qudits*, described by vectors from $\mathbb{C}^m$. Measuring such a qudit would give a result from an alphabet with m symbols. In general, this alphabet could be any finite Abelian group, but we will only consider the case where the alphabet is a finite field. The m vectors $|x\rangle$, $x \in$ GF(m), form an orthonormal basis of $\mathbb{C}^m$.

An error operator that can affect a single qudit is represented by a complex *unitary* $m \times m$ matrix, i.e., a matrix U such that $UU^\dagger = I$, where $\dagger$ means conjugate transpose. A state of n qudits is represented by a vector from $\mathbb{C}^{m^n} = \mathbb{C}^m \otimes \cdots \otimes \mathbb{C}^m$. Assuming that errors act independently on each qubit, this state is affected by error operators described by n-fold tensor products of unitary $m \times m$ matrices. In the case of qubits ($m = 2$), we only need to consider errors from the *Pauli group*,

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad Y = iXZ \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

due to the fact that these matrices form a basis of all unitary 2×2 matrices. The error X is called a *bit-flip*, since $X|x\rangle = |x+1\rangle$. The error Z is known as a *phase-flip*, since $Z|x\rangle = (-1)^x|x\rangle$. For general qudits that take their values from $\mathrm{GF}(m)$, we consider the *generalized Pauli group*, $\mathcal{P}_m$, also known as the *discrete Heisenberg-Weyl group*. When our alphabet is a finite field, we must have $m = p^r$, where p is a prime and $r \geq 1$. The errors contained in the generalized Pauli group are *shift errors*, $X(a)|x\rangle = |x+a\rangle$, and *phase errors*, $Z(b)|x\rangle = \omega^{\mathrm{Tr}_{m/p}(bx)}|x\rangle$, where $a, b \in \mathrm{GF}(m)$, ω is a complex p-th root of unity, and $\mathrm{Tr}_{m/p} : \mathrm{GF}(m) \to \mathrm{GF}(p)$ is the trace function, $\mathrm{Tr}_{m/p}(c) = \sum_{i=0}^{r-1} c^{p^i}$. If $m = p$ is a prime, i.e., $r = 1$, the generalized Pauli group is generated by

$$\left\langle X(1) = \begin{pmatrix} 0 & & & \\ \vdots & & I_{m-1} & \\ 0 & & & \\ 1 & 0 & \cdots & 0 \end{pmatrix}, \quad Z(1) = \begin{pmatrix} 1 & & & & 0 \\ & \omega & & & \\ & & \omega^2 & & \\ & & & \ddots & \\ 0 & & & & \omega^{n-1} \end{pmatrix} \right\rangle,$$

where ω is a complex p-th root of unity, and I is the identity matrix of specified dimension.[1] The operators $X(a)$ and $Z(b)$ are obtained by taking the a-th and b-th powers of $X(1)$ and $Z(1)$, respectively. Even if m is not prime, we can still define qudits that take values from the cyclic group $\mathbb{Z}_m$, and use the same error operators as defined above. However, when m is a prime power, we get much better codes by using a finite field as our alphabet. When we work with qudits that take values from

[1] The set of generators also contains the scalar ω, except for the case $m = 2$, where it contains i, a 4-th root of unity. This overall phase factor can be ignored for our purposes.

GF(p^r), where $r > 1$, we use the error group $\{\bigotimes_{i=0}^{r} E_i \mid E_i \in \mathcal{P}_p\}$ [3], i.e., the operators are r-fold tensor products of Pauli matrices from the group $\mathcal{P}_p$. The error bases that we use are examples of *nice error bases* [16].

Quantum codes are designed to add redundancy in order to protect quantum states against errors due to interference from the environment. A code of *length* n and *dimension* k adds redundancy by encoding k qudits using n qudits. One type of code that exploits the fact that the generalized Pauli group forms a basis for all possible errors is the *stabilizer code* [17]. A *stabilizer* is an Abelian group generated by a set of $n - k$ commuting error operators. An error is detected by measuring the eigenvalues of these operators. If a state is a valid codeword that has not been affected by error, we will observe the eigenvalue $+1$ for all operators. The quantum code, i.e., the set of all valid codewords, is therefore a joint eigenspace of the stabilizer. If there is a detectable error, some eigenvalues would be different from $+1$, due to the commutativity properties of the generalized Pauli matrices. A stabilizer generated by a set of n error operators defines a zero-dimensional quantum code, also known as a *stabilizer state*.[2] The *minimum distance* of a zero-dimensional stabilizer code is simply the minimum *weight* of all error operators in the stabilizer. The weight of an error operator is the number of $m \times m$ tensor components that are different from the identity matrix. A quantum code of length n, dimension k, and minimum distance d, over the alphabet GF(m), is denoted an $[[n, k, d]]_m$ code. Stabilizer states are therefore $[[n, 0, d]]_m$ codes. If the minimum distance is high, the stabilizer state is robust against error, which indicates that it is highly *entangled*. Entangled quantum states have many potential applications, for instance in cryptographic protocols, or as *graph states* [18] which can be used as a resource for quantum computations. In the next section we will also see that zero-dimensional stabilizer codes correspond to an interesting class of classical codes, known as *self-dual additive codes*.

[2]Stabilizer states could also be called one-dimensional quantum codes, since they are one-dimensional Hilbert subspaces. We use the term dimension to mean the number of qudits the code can encode.

Example 1. A $[[4,0,3]]_3$ stabilizer state is obtained from the stabilizer generated by the following error operators.

$$\begin{array}{llll} X(1) & \otimes X(1)Z(2) & \otimes I & \otimes X(1), \\ X(1)Z(1) & \otimes X(2) & \otimes X(1)Z(1) & \otimes X(1), \\ I & \otimes X(2)Z(2) & \otimes X(1)Z(1) & \otimes Z(2), \\ X(1) & \otimes X(2)Z(2) & \otimes X(2) & \otimes X(2)Z(2). \end{array}$$

3 Self-Dual Additive Codes

We can represent a stabilizer state over GF(m) by an $n \times 2n$ matrix $(A \mid B)$ [4]. The submatrix A represents shift errors, such that $A_{(i,j)} = a$ if $X(a)$ occurs in the j-th tensor component of the i-th error operator in the set of generators. Similarly, the submatrix B represents phase errors.

Example 2. The matrix corresponding to the stabilizer state in Example 1 is

$$(A \mid B) = \left(\begin{array}{cccc|cccc} 1 & 1 & 0 & 1 & 0 & 2 & 0 & 0 \\ 1 & 2 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 2 & 1 & 0 & 0 & 2 & 1 & 2 \\ 1 & 2 & 2 & 2 & 0 & 2 & 0 & 2 \end{array}\right).$$

The matrix $(A \mid B)$ generates a code $\mathcal{C}$, and this code is a representation of a stabilizer state. The fact that a stabilizer is an Abelian group translates into the requirement that $\mathcal{C}$ must be *self-dual* with respect to a *symplectic inner product*, i.e.,

$$(a \mid b) * (a' \mid b') = \mathrm{Tr}_{m/p}(b \cdot a' - b' \cdot a) = 0, \quad \forall (a \mid b), (a' \mid b') \in \mathcal{C}. \quad (1)$$

We define the *symplectic weight* of a codeword $(a \mid b) \in \mathcal{C}$ as the number of positions i where both a_i and b_i are nonzero. (This is the same as the weight of the corresponding Pauli error operator.)

We can also map the linear code of length $2n$ defined above to an additive code over GF(m^2) of length n. The representation of binary stabilizer codes as self-dual additive codes over GF(4) was first demonstrated by Calderbank et al. [1], and the generalization to qudits was completed by Ketkar et al. [3]. An *additive* code, $\mathcal{C}$, over GF(m^2) of length n is defined as an GF(m)-linear subgroup of GF$(m^2)^n$. The code $\mathcal{C}$ contains m^n codewords, and can be defined by an $n \times n$ generator matrix, C, with entries from GF(m^2), such that any GF(m)-linear

combination of rows from C is a codeword.[3] To get from the stabilizer representation $(A \mid B)$ to the generator matrix C, we simply take $C = A + \omega B$, where ω is a primitive element of $\mathrm{GF}(m^2)$. The code $\mathcal{C}$ will be self-dual, $\mathcal{C} = \mathcal{C}^\perp$, where the dual is defined with respect to the *Hermitian trace inner product*, $\mathcal{C}^\perp = \{\boldsymbol{u} \in \mathrm{GF}(m^2)^n \mid \boldsymbol{u} * \boldsymbol{c} = 0 \text{ for all } \boldsymbol{c} \in \mathcal{C}\}$. When $m = p$ is prime, the Hermitian trace inner product of two vectors over $\mathrm{GF}(p^2)$ of length n, $\boldsymbol{u} = (u_1, u_2, \ldots, u_n)$ and $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$, is given by

$$\boldsymbol{u} * \boldsymbol{v} = \mathrm{Tr}_{p^2/p}(\boldsymbol{u} \cdot \boldsymbol{v}^p) = \boldsymbol{u} \cdot \boldsymbol{v}^p - \boldsymbol{u}^p \cdot \boldsymbol{v} = \sum_{i=1}^{n} (u_i v_i^p - u_i^p v_i), \qquad (2)$$

When $m = p^r$ is not a prime, we use a modification of the Hermitian trace inner product [3],

$$\boldsymbol{u} * \boldsymbol{v} = \mathrm{Tr}_{m/p}\left(\frac{\boldsymbol{u} \cdot \boldsymbol{v}^m - \boldsymbol{u}^m \cdot \boldsymbol{v}}{\omega - \omega^m}\right), \qquad (3)$$

where ω is a primitive element of $\mathrm{GF}(m^2)$.

The *Hamming weight* of a codeword $\boldsymbol{u} \in \mathcal{C}$, denoted $\mathrm{wt}(\boldsymbol{u})$, is the number of nonzero components of $\boldsymbol{u}$. The *Hamming distance* between $\boldsymbol{u}$ and $\boldsymbol{v}$ is $\mathrm{wt}(\boldsymbol{u} - \boldsymbol{v})$. The *minimum distance* of the code $\mathcal{C}$ is the minimal Hamming distance between any two distinct codewords of $\mathcal{C}$. Since $\mathcal{C}$ is an additive code, the minimum distance is also given by the smallest nonzero weight of any codeword in $\mathcal{C}$. A code over $\mathrm{GF}(m^2)$ with minimum distance d is called an (n, m^n, d) code. The *weight distribution* of the code $\mathcal{C}$ is the sequence $(A_0, A_1, \ldots, A_n)$, where A_i is the number of codewords of weight i. For an additive code over $\mathrm{GF}(m^2)$, all A_i must be divisible by $m - 1$.

Example 3. The stabilizer state in Example 1 corresponds to the following generator matrix of a self-dual additive $(4, 3^4, 3)$ code.

$$C = \begin{pmatrix} 1 & 1+2\omega & 0 & 1 \\ 1+\omega & 2 & 1+\omega & 1 \\ 0 & 2+2\omega & 1+\omega & 2\omega \\ 1 & 2+2\omega & 2 & 2+2\omega \end{pmatrix}.$$

We define two self-dual additive codes, $\mathcal{C}$ and $\mathcal{C}'$ over $\mathrm{GF}(m^2)$, to be *equivalent* if the codewords of $\mathcal{C}$ can be mapped onto the codewords of $\mathcal{C}'$

[3] For codes over $\mathrm{GF}(4)$, each codeword is a sum of rows of the generator matrix, hence the name "additive code". However, the code is still called additive in the general case [19].

by certain maps that preserve self-duality. A permutation of coordinates, or columns of a generator matrix, is such a map. Other operations can also be applied to the coordinates of $\mathcal{C}$. Let each element $a + \omega b \in \mathrm{GF}(m^2)$ be represented as $\binom{a}{b} \in \mathrm{GF}(m)^2$. We can then premultiply this element by a 2×2 matrix. (We could equivalently have applied transformations to pairwise columns of the $2n \times n$ matrix $(A|B)$.) It was shown by Rains [2] that by applying matrices from the *symplectic group* $\mathrm{Sp}_2(m)$ to each coordinate, we get an equivalent code. (This group contains all 2×2 matrices with elements in $\mathrm{GF}(m)$ and determinant 1.) For self-dual additive codes over $\mathrm{GF}(4)$, these symplectic operations can be represented more simply as multiplication by nonzero elements from $\mathrm{GF}(4)$ and conjugation of coordinates. (Conjugation of elements in $\mathrm{GF}(p^2)$ maps x to x^p.) Combined, there are 6 possible transformations that are equivalent to the 6 permutations of the elements $\{1, \omega, \omega^2\}$ in the coordinate. The corresponding symplectic group is

$$\mathrm{Sp}_2(2) = \left\langle A_1 = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, A_2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \right\rangle,$$

where A_1 represents multiplication by ω and A_2 represents conjugation. Including coordinate permutations, there are a total of $6^n n!$ maps for a code of length n.

For codes over $\mathrm{GF}(9)$, we observe that $\mathrm{Sp}_2(3)$ is a group of order 24 generated by

$$\mathrm{Sp}_2(3) = \left\langle A_1 = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}, A_2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \right\rangle,$$

where A_1 represents multiplication by ω^2 and A_2 represents the map $a + \omega b \mapsto a + b + \omega b$. It follows that we are allowed to multiply a coordinate by $x \in \mathrm{GF}(9)$ only if $x\overline{x} = 1$. However, if we also conjugate the coordinate, we may multiply by $x \in \mathrm{GF}(9)$ where $x\overline{x} = 2$. Note that conjugation on its own is not allowed. The 8 operations just described may be combined with the operations represented by A_2 and A_2^2 to give a total of 24 operations. In all there are $24^n n!$ maps that take a self-dual additive code over $\mathrm{GF}(9)$ to an equivalent code. In general, for codes over $\mathrm{GF}(m^2)$, the number of maps is $|\mathrm{Sp}_2(m)|^n n!$.

A transformation that maps $\mathcal{C}$ to itself is called an *automorphism* of $\mathcal{C}$. All automorphisms of $\mathcal{C}$ make up an *automorphism group*, denoted $\mathrm{Aut}(\mathcal{C})$. The number of distinct codes equivalent to a self-dual additive code over $\mathrm{GF}(m^2)$, $\mathcal{C}$, is then given by $\frac{|\mathrm{Sp}_2(m)|^n n!}{|\mathrm{Aut}(\mathcal{C})|}$. The *equivalence class*

of $\mathcal{C}$ contains all codes that are equivalent to $\mathcal{C}$. By summing the sizes of all equivalence classes of codes of length n, we find the total number of distinct codes of length n, denoted T_n. The number T_n is also given by a *mass formula*. The mass formula for self-dual additive codes over GF(4) was found by Höhn [20]. This result is easily generalized to GF(m^2).

Theorem 4.

$$T_n = \prod_{i=1}^{n}(m^i + 1) = \sum_{j=1}^{t_n} \frac{|\operatorname{Sp}_2(m)|^n n!}{|\operatorname{Aut}(\mathcal{C}_j)|}, \tag{4}$$

where t_n is the number of equivalence classes of codes of length n, and $\mathcal{C}_j$ is a representative from each equivalence class.

Proof. Let $M(n,k)$ be the total number of self-orthogonal (n, m^k) codes. One such code, $\mathcal{C}$, can be extended to a self-orthogonal (n, m^{k+1}) code in $m^{2(n-k)} - 1$ ways by adding an extra codeword from $C^{\perp}$. Each (n, m^{k+1}) code can be obtained in this way from $m^{2(k+1)} - 1$ different (n, m^k) codes. It follows that

$$M(n, k+1) = M(n,k)\frac{m^{2(n-k)} - 1}{m^{2(k+1)} - 1}.$$

Starting with $M(n, 0) = 1$, the recursion gives us the number of self-dual (n, m^n) codes,

$$M(n,n) = \prod_{i=0}^{n-1} \frac{m^{2(n-k)} - 1}{m^{2(k+1)} - 1} = \prod_{i=1}^{n}(m^i + 1).$$

□

By assuming that all codes of length n have a trivial automorphism group, we get the following lower bound on t_n, the total number of inequivalent codes. Note that when n is large, most codes have a trivial automorphism group, so the tightness of the bound increases with n. Also note that this bound is much tighter than a bound that was derived from results in graph theory by Bahramgiri and Beigi [15].

Theorem 5.

$$t_n \geq \left\lceil \frac{c\prod_{i=1}^{n}(m^i + 1)}{|\operatorname{Sp}_2(m)|^n n!} \right\rceil, \tag{5}$$

where $c = 1$ if m is even, and $c = 2$ if m is odd.

Proof. When m is even, the trivial automorphism group includes only the identity permutation, and the result follows from Theorem 4. When $m = p^r$ is odd, where p is a prime, the trivial automorphism group also contains the transformation that applies the symplectic operation $\begin{pmatrix} p-1 & 0 \\ 0 & p-1 \end{pmatrix}$ to all coordinates. This operation is equivalent to multiplying each codeword by $p-1$, and will therefore map an additive code to itself. □

It follows from the *quantum singleton bound* [2, 21] that any self-dual additive code must satisfy $2d \leq n+2$. A tighter bound for codes over GF(4) was given by Calderbank et al. [1]. Codes that satisfy the singleton bound with equality are known as *maximum distance separable (MDS) codes*. MDS codes must have even length, and MDS codes of length 2 are trivial and exist for all alphabets. The only non-trivial MDS code over GF(4) is the $(6, 2^6, 4)$ *Hexacode*. Ketkar et al. [3, Thm. 63] proved that a self-dual additive (n, m^n, d) MDS code must satisfy $n \leq m^2 + d - 2 \leq 2m^2 - 2$. If the famous MDS conjecture holds, then $n \leq m^2 + 1$, or $n \leq m^2 + 2$ when m is even and $d = 4$ or $d = m^2$. Grassl, Rötteler, and Beth [22] showed that MDS codes of length $n \leq m + 1$ always exist. After we have defined the concept of graph codes in Section 4, we give the results of a search of circulant graph codes in Section 7, which shows what minimum distances can be expected for self-dual additive codes of short length over various alphabets.

Self-dual *linear* codes over GF(m^2) are a subset of the self-dual additive codes. Only additive codes that satisfy certain constraints can be linear. Such constraints for codes over GF(4) were described by Van den Nest [11] and by Glynn et al. [13], and can be generalized to other alphabets. An obvious constraint is that all coefficients of the weight enumerator of a linear code must be divisible by $m^2 - 1$, whereas for an additive code they need only be divisible by $m - 1$.

4 Correspondence to Weighted Graphs

A *graph* is a pair $G = (V, E)$ where V is a set of *vertices* and $E \subseteq V \times V$ is a set of *edges*. Let an *m-weighted graph* be a triple $G = (V, E, W)$ where W is a set of weights from GF(m). Each edge has an associated non-zero weight. (An edge with weight zero is the same as a non-edge.) An m-weighted graph with n vertices can be represented by an

$n \times n$ adjacency matrix Γ, where the element $\gamma_{ij} = W(\{i,j\})$ if $\{i,j\} \in E$, and $\gamma_{ij} = 0$ otherwise. We will only consider *simple undirected* graphs whose adjacency matrices are symmetric with all diagonal elements being 0. The *neighbourhood* of $v \in V$, denoted $N_v \subset V$, is the set of vertices connected to v by an edge. The number of vertices adjacent to v, $|N_v|$, is called the *degree* of v. The *induced subgraph* of G on $U \subseteq V$ contains vertices U and all edges from E whose endpoints are both in U. The *complement* of a non-weighted graph G is found by replacing E with $V \times V - E$, i.e., the edges in E are changed to non-edges, and the non-edges to edges. Two graphs $G = (V, E)$ and $G' = (V, E')$ are *isomorphic* if and only if there exists a permutation π of V such that $\{u,v\} \in E \iff \{\pi(u), \pi(v)\} \in E'$. We also require that weights are preserved, i.e., $W_{\{u,v\}} = W_{\{\pi(u),\pi(v)\}}$. A *path* is a sequence of vertices, $(v_1, v_2, \ldots, v_i)$, such that $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{i-1}, v_i\} \in E$. A graph is *connected* if there is a path from any vertex to any other vertex in the graph. A *complete graph* is a graph where all pairs of vertices are connected by an edge. A *clique* is a complete subgraph.

Definition 6. A *graph code* is an additive code over $\mathrm{GF}(m^2)$ that has a generator matrix of the form $C = \Gamma + \omega I$, where I is the identity matrix, ω is a primitive element of $\mathrm{GF}(m^2)$, and Γ is the adjacency matrix of a simple undirected m-weighted graph.

Theorem 7. *Every self-dual additive code over* $\mathrm{GF}(m^2)$ *is equivalent to a graph code.*

Proof. The generator matrix, C, of a self-dual additive code over $\mathrm{GF}(m^2)$ corresponds to an $n \times 2n$ matrix $(A|B)$ with elements from $\mathrm{GF}(m)$, such that $C = A + \omega B$. We must prove that an equivalent code is generated by $(\Gamma|I)$, where I is the identity matrix and Γ is the adjacency matrix of a simple undirected m-weighted graph. A basis change can be accomplished by $(A'|B') = M(A|B)$, where M is an $n \times n$ invertible matrix with elements from $\mathrm{GF}(m)$. If B has full rank, we can find a matrix B^* such that $BB^* = cI$, for some constant c. The solution is then simple, since $B^*(A|B) = (\Gamma'|cI)$. We obtain $(\Gamma|I)$ after changing the diagonal elements of Γ' and cI to 0 and 1, respectively, by appropriate symplectic transformations. Any two rows of $(\Gamma|I)$ will be orthogonal with respect to the symplectic inner product, which means that $\Gamma I^{\mathrm{T}} - I\Gamma^{\mathrm{T}} = 0$, and it follows that Γ will always be a symmetric

matrix. In the case where B has rank $k < n$, we can perform a basis change to get

$$(A'|B') = \left(\begin{array}{c|c} A_1 & B_1 \\ A_2 & \mathbf{0} \end{array}\right),$$

where B_1 is a $k \times n$ matrix with full rank, and A_1 also has size $k \times n$. Since the row-space of $(A'|B')$ defines a self-dual code, and B' contains an all-zero row, it must be true that $A_2 B_1^{\mathrm{T}} = \mathbf{0}$. A_2 must have full rank, and the row space of B_1 must be the orthogonal complement of the row space of A_2.

We assume that $B_1 = (B_{11}|B_{12})$ where B_{11} is a $k \times k$ invertible matrix. We also write $A_2 = (A_{21}|A_{22})$ where A_{22} has size $(n-k) \times (n-k)$. Assume that there exists an $\boldsymbol{x} \in \mathrm{GF}(m)^{n-k}$ such that $A_{22}\boldsymbol{x}^{\mathrm{T}} = 0$. Then the vector $\boldsymbol{v} = (0, \ldots, 0, \boldsymbol{x})$ of length n satisfies $A_2\boldsymbol{v}^{\mathrm{T}} = 0$. Since the row space of B_1 is the orthogonal complement of the row space of A_2, we can write $\boldsymbol{v} = \boldsymbol{y}B_1$ for some $\boldsymbol{y} \in \mathrm{GF}(m)^k$. We see that $\boldsymbol{y}B_{11} = 0$, and since B_{11} has full rank, it must therefore be true that $\boldsymbol{y} = 0$. This means that $\boldsymbol{x} = 0$, which proves that A_{22} is an invertible matrix.

Two of the symplectic operations that we can apply to columns of a generator matrix are $\begin{pmatrix} 0 & m-1 \\ 1 & 0 \end{pmatrix}$ and $\begin{pmatrix} 0 & 1 \\ m-1 & 0 \end{pmatrix}$. This means that we can interchange column i of A' and column i of B' if we also multiply one of the columns by $m-1$. In this way we swap the i-th columns of A' and B' for $k < i \leq n$ to get $(A''|B'')$. Since B_{11} and A_{22} are invertible, B'' must also be an invertible matrix. We then find $B''^{-1}(A''|B'') = (\Gamma|I)$, and set all diagonal elements of Γ to 0 by symplectic transformations. □

Example 8. The matrix from Example 2 can be transformed into the following matrix, using the method given in the proof of Theorem 7.

$$(\Gamma \mid I) = \left(\begin{array}{cccc|cccc} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 & 0 & 1 & 0 \\ 0 & 1 & 2 & 0 & 0 & 0 & 0 & 1 \end{array}\right).$$

This means that the stabilizer state from Example 1 is equivalent to the graph code generated by $C = \Gamma + \omega I$.

Note that Theorem 7 is a generalization of the same theorem for codes over GF(4) [14], which was proved by Van den Nest et al. [12]. The fact that stabilizer codes can be represented by graphs was also

shown by Schlingemann and Werner [9] and by Grassl, Klappenecker, and Rötteler [6].

We have seen that every m-weighted graph represents a self-dual additive code over GF(m), and that every self-dual additive code over GF(m) can be represented by an m-weighted graph. It follows that we can, without loss of generality, restrict our study to codes with generator matrices of the form $\Gamma + \omega I$, where Γ is an adjacency matrix of an unlabeled simple undirected m-weighed graph.

5 Graph Equivalence and Code Equivalence

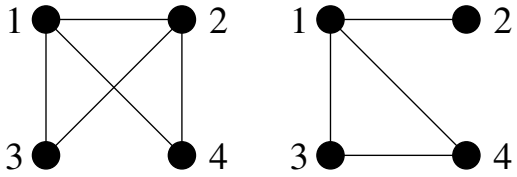
Swapping vertex i and vertex j of a graph with adjacency matrix Γ can be accomplished by exchanging column i and column j of Γ and then exchanging row i and row j of Γ. We call the resulting matrix Γ'. Exactly the same column and row operations map $\Gamma + \omega I$ to $\Gamma' + \omega I$, which are generator matrices for equivalent codes. It follows that two codes are equivalent if their corresponding graphs are isomorphic. However, the symplectic transformations that map a code to an equivalent code do not in general produce isomorphic graphs, but we will see that they can be described as graph operations.

It is known that two self-dual additive codes over GF(4) are equivalent if and only if their corresponding graphs are equivalent, up to isomorphism, with respect to a sequence of *local complementations* [10–13]. We have previously used this fact to devise a graph-based algorithm with which we classified all self-dual additive codes over GF(4) of length up to 12 [14].

Definition 9 ([10])**.** Given a graph $G = (V, E)$ and a vertex $v \in V$, let $N_v \subset V$ be the neighbourhood of v. *Local complementation* (LC) on v transforms G into $G * v$ by replacing the induced subgraph of G on N_v by its complement. (Fig. 1)

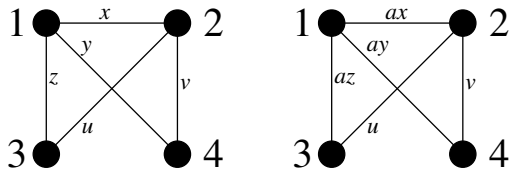
Theorem 10 ([10–13])**.** *Two self-dual additive codes over* GF(4)*, $\mathcal{C}$ and $\mathcal{C}'$, with graph representations G and G', are equivalent if and only if there is a finite sequence of not necessarily distinct vertices $(v_1, v_2, \ldots, v_i)$, such that $G * v_1 * v_2 * \cdots * v_i$ is isomorphic to G'.*

The LC operation can be generalized to weighted graphs, and it was first shown by Bahramgiri and Beigi [15] that the equivalence of



(a) *The Graph G* **(b)** *The Graph* $G * 1$

Fig. 1: *Example of Local Complementation*



(a) *The Graph G* **(b)** *The Graph* $G \circ_a 1$

Fig. 2: *Example of Weight Shifting*

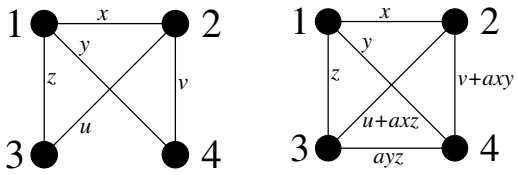
nonbinary stabilizer states over GF(m), i.e., self-dual additive codes over GF(m^2), can be described in terms of graph operations.[4]

Definition 11 ([15])**.** Given an m-weighted graph $G = (V, E, W)$ and a vertex $v \in V$, *weight shifting* on v by $a \in \mathrm{GF}(m)$ transforms G into $G \circ_a v$ by multiplying the weight of each edge incident on v by a. (Fig. 2)

Definition 12 ([15])**.** Given an m-weighted graph $G = (V, E, W)$ and a vertex $v \in V$, *generalized local complementation* on v by $a \in \mathrm{GF}(m)$ transforms G into $G *_a v$. Let Γ and Γ' be the adjacency matrices of G and $G *_a v$, respectively. Then $\Gamma'_{ij} = \Gamma_{ij} + a\Gamma_{vi}\Gamma_{vj}$, for all $i \neq j$, and $\Gamma'_{ii} = 0$ for all i. (Fig. 3)

Theorem 13 ([15])**.** *Two self-dual additive codes over* GF(m^2), $\mathcal{C}$ *and* $\mathcal{C}'$*, with graph representations* G *and* G'*, are equivalent if and only if we get a graph isomorphic to* G' *by applying some finite sequence of weight shifts and generalized local complementations to* G*.*

[4] Bahramgiri and Beigi [15] only state their theorem for GF(m) where m is prime, but the result holds for any finite field, as their proof does not depend on m being prime.



(a) *The Graph G* **(b)** *The Graph* $G *_a 1$

Fig. 3: *Example of Generalized Local Complementation*

A proof of Theorem 13 was given by by Bahramgiri and Beigi [15], as a generalization of the proof given by Van den Nest et al. [12] for self-dual additive codes over GF(4).

Theorem 14. *The minimum distance of a self-dual additive* (n, m^n, d) *code is equal to* $\delta + 1$*, where* δ *is the minimum vertex degree over all graphs in the associated generalized LC orbit.*

Proof. A vertex with degree $d - 1$ in the LC orbit corresponds to a codeword of weight d, and we will now show that such a vertex always exists. Choose any graph representation of the code and let $G = (\Gamma \mid I)$ be the corresponding generator matrix. Find a codeword c of weight d generated by G. Let the i-th row of G be one of the rows that c is linearly dependent on. Apply symplectic transformations to the coordinates of the code such that c is mapped to c' with 1 in coordinate $n + i$, and with 0 in all other of the last n coordinates. Since we do not care about changes in the corresponding first n coordinates, there will always be transformations that achieve this. Apply the same transformations to the columns of G, and then replace the i-th row with c', to get G'. Note that the right half of G' still has full rank, so we can transform G' into a matrix of the form $(\Gamma' \mid I)$ by Gaussian elimination, where the symplectic weight of the i-th row is d. Finally, we set all diagonal elements of Γ' to zero by appropriate symplectic transformations. Vertex i of the graph with adjacency matrix Γ' has degree $d - 1$. □

6 Classification

Definition 15. The *LC orbit* of a weighted graph G is the set of all non-isomorphic graphs that can be obtained by performing any sequence of weight shifts and generalized LC operations on G.

It follows from Theorem 13 that two self-dual additive codes over $\mathrm{GF}(m^2)$ are equivalent if and only if their graph representations are in the same LC orbit. The LC orbit of a graph can easily be generated by a recursive algorithm. We have used the program *nauty* [23] to check for graph isomorphism.

Let $\boldsymbol{G}_{n,m}$ be the set of all non-isomorphic simple undirected connected m-weighted graphs on n vertices. Note that connected graphs correspond to *indecomposable* codes. A code is decomposable if it can be written as the *direct sum* of two smaller codes. For example, let $\mathcal{C}$ be an (n, m^n, d) code and $\mathcal{C}'$ an $(n', m^{n'}, d')$ code. The direct sum, $\mathcal{C} \oplus \mathcal{C}' = \{u||v \mid u \in \mathcal{C}, v \in \mathcal{C}'\}$, where $||$ means concatenation, is an $(n+n', m^{n+n'}, \min\{d, d'\})$ code. It follows that all decomposable codes of length n can be classified easily once all indecomposable codes of length less than n are known.

The set of all distinct LC orbits of connected m-weighted graphs on n vertices is a partitioning of $\boldsymbol{G}_{n,m}$ into $i_{n,m}$ disjoint sets. $i_{n,m}$ is also the number of indecomposable self-dual additive codes over $\mathrm{GF}(m^2)$ of length n, up to equivalence. Let $\boldsymbol{L}_{n,m}$ be a set containing one representative from each LC orbit of connected m-weighted graphs on n vertices. The simplest algorithm for finding such sets of representatives is to start with the set $\boldsymbol{G}_{n,m}$ and generate LC orbits of its members until we have a partitioning of $\boldsymbol{G}_{n,m}$. The following more efficient technique is based on a method described by Glynn et al. [13]. Let the $m^n - 1$ *extensions* of an m-weighted graph on n vertices be formed by adding a new vertex and joining it to all possible combinations of at least one of the old vertices, using all possible combinations of edge weights. The set $\boldsymbol{E}_{n,m}$, containing $i_{n-1,m}(m^{n-1} - 1)$ graphs, is formed by making all possible extensions of all graphs in $\boldsymbol{L}_{n-1,m}$.

Theorem 16. *$\boldsymbol{L}_{n,m} \subset \boldsymbol{E}_{n,m}$, i.e., the set $\boldsymbol{E}_{n,m}$ will contain at least one representative from each LC orbit of connected graphs on n vertices.*

Proof. Let $G = (V, E, W) \in \boldsymbol{G}_{n,m}$, and choose any subset $U \subset V$ of $n-1$ vertices. By doing weight shifts and generalized LC operations on vertices in U, we can transform the induced subgraph of G on U into one of the graphs in $\boldsymbol{L}_{n-1,m}$ that were extended when $\boldsymbol{E}_{n,m}$ was constructed. It follows that for all $G \in \boldsymbol{G}_{n,m}$, some graph in the LC orbit of G must be part of $\boldsymbol{E}_{n,m}$. □

The set $\boldsymbol{E}_{n,m}$ will be much smaller than $\boldsymbol{G}_{n,m}$, so it will be more efficient to search for a set of LC orbit representatives within $\boldsymbol{E}_{n,m}$.

Another fact that simplifies our classification algorithm is that weight shifting and generalized local complementation commute. This means that to generate the LC orbit of a weighted graph, we may first generate the orbit with respect to generalized local complementation only, and then apply weight shifting to the resulting set of graphs.

Using the described techniques, we were able to classify all self-dual additive codes over GF(9), GF(16), and GF(25) up to lengths 8, 6, and 6, respectively. Table 1 gives the values of $i_{n,m}$, the number of distinct LC orbits of connected m-weighted graphs on n vertices, which is also the number of inequivalent indecomposable self-dual additive codes over GF(m^2) of length n. The total number of inequivalent codes of length n, t_n, is shown in Table 2 together with lower bounds derived from Theorem 5. The numbers t_n are easily derived from the numbers i_n by using the *Euler transform* [24],

$$\begin{aligned} c_n &= \sum_{d|n} d i_d \\ t_1 &= c_1 \\ t_n &= \frac{1}{n}\left(c_n + \sum_{k=1}^{n-1} c_k t_{n-k}\right). \end{aligned}$$

Tables 3, 4, and 5 list by minimum distance the numbers of indecomposable codes over GF(9), GF(16), and GF(25). A database containing one representative from each equivalence class is available at `http://www.ii.uib.no/~larsed/nonbinary/`.

Note that applying the graph extension technique described previously is equivalent to *lengthening* [25] a self-dual additive code. Given an (n, m^n, d) code, we add a row and column to its generator matrix to obtain an $(n+1, m^{n+1}, d')$ code, where $d' \leq d+1$. If follows that given a classification of all codes of length n and distance d, we can classify all codes of length $n+1$ and distance $d+1$. All length 8 codes over GF(9) have been classified as described above. By extending the 77 $(8, 3^8, 4)$ codes, we found 4 $(9, 3^9, 5)$ codes, and from those we obtained a single $(10, 3^{10}, 6)$ code. Assuming that the MDS conjecture holds, there are no self-dual additive MDS codes over GF(9) with length above 10. This would mean that the three MDS codes we have found, with parameters $(4, 3^4, 3)$, $(6, 3^6, 4)$, and $(10, 3^{10}, 6)$, are the only self-dual additive MDS codes over GF(9).

Table 1: *Number ($i_{n,m}$) of Indecomposable Codes of Length n over* GF(m^2)

n	$i_{n,2}$	$i_{n,3}$	$i_{n,4}$	$i_{n,5}$
1	1	1	1	1
2	1	1	1	1
3	1	1	3	3
4	2	3	6	7
5	4	5	25	38
6	11	21		
7	26	73		
8	101	659		
9	440			
10	3132			
11	40 457			
12	1 274 068			

Table 2: *Total Number ($t_{n,m}$) of Codes of Length n over* GF(m^2)

n	$t_{n,2}$	$t_{n,3}$	$t_{n,4}$	$t_{n,5}$
1	1	1	1	1
2	2	2	2	2
3	3	3	5	5
4	6	7	12	13
5	11	13	40	54
6	26	39	?	?
7	59	121	?	?
8	182	817	$\geq$ *946*	$\geq$ *21 161*
9	675	$\geq$ *9 646*	$\geq$ *458 993*	$\geq$ *38 267 406*
10	3990	$\geq$ *2 373 100*		
11	45 144			
12	1 323 363			
13	$\geq$ *72 573 549*			

Table 3: *Number of Indecomposable Codes of Length n and Distance d over* GF(9)

$d\backslash n$	2	3	4	5	6	7	8	9	10
2	1	1	2	4	15	51	388	?	?
3			1	1	5	20	194	?	?
4					1	2	77	?	?
5								4	?
6									1
All	1	1	3	5	21	73	659	?	?

Table 4: *Number of Indecomposable Codes of Length n and Distance d over* GF(16)

$d\backslash n$	2	3	4	5	6
2	1	1	2	4	16
3			1	2	6
4					3
5					
6					
All	1	1	3	6	25

Table 5: *Number of Indecomposable Codes of Length n and Distance d over* GF(25)

$d\backslash n$	2	3	4	5	6
2	1	1	2	4	21
3			1	3	11
4					6
5					
6					
All	1	1	3	7	38

7 Circulant Graph Codes

It is clearly infeasible to study all self-dual additive codes of lengths much higher than those classified in the previous section. We therefore restrict our search space to the $m^{\lceil \frac{n-1}{2} \rceil}$ codes over $\mathrm{GF}(m^2)$ of length n corresponding to graphs with *circulant* adjacency matrices. A matrix is circulant if the i-th row is equal to the first row, cyclically shifted i times to the right. We have performed an exhaustive search of such graphs, the result of which is summarized in Table 6. This table shows the highest found minimum distance of self-dual additive codes over various alphabets. A code with the given distance has been found in our search, except for the cases marked $*$, where the code is obtained in some other way and does not have any circulant graph representation,[5] and cases marked s, which are not circulant, but obtained by a trivial *shortening* [25] of a longer circulant code. Distances printed in bold font are optimal according to the quantum singleton bound. If n is even and the quantum singleton bound is satisfied with equality, we have an MDS code.

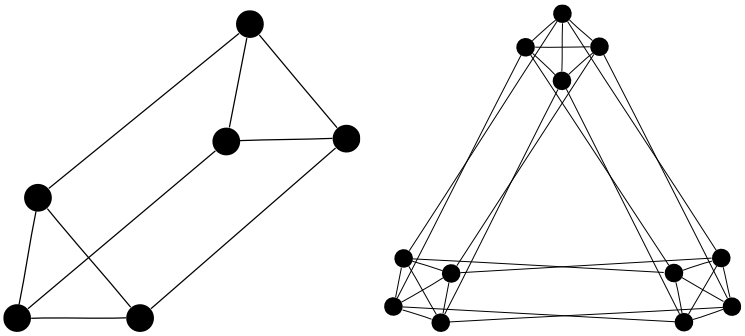
As mentioned in the introduction, stabilizer codes can be defined over any Abelian group, not only finite fields. For comparison, we also generated circulant codes over $\mathbb{Z}_4^2$. As expected, the minimum distance of these codes are much worse than for codes over $\mathrm{GF}(16)$. We found a $(7, 4^7, 4)$-code over $\mathbb{Z}_4^2$, but for all other lengths up to 16, the best distance was equal to the best distance of codes over $\mathrm{GF}(4)$ of the same length.

Gulliver and Kim [26] performed a computer search of circulant self-dual additive codes over $\mathrm{GF}(4)$ of length up to 30. Their search was not restricted to graph codes, so our search space is a subset of theirs. It is interesting to note that for every length, the highest minimum distance found was the same in both searches. This suggests that the circulant graph code construction can produce codes as strong as the more general circulant code construction. Besides a smaller search space, the special form of the generator matrix of a graph code makes it easier to find the minimum distance, since any codeword obtained as a linear combination of i rows of the generator matrix must have weight at least i. If, for example, we want to determine whether a code has minimum distance at least d, we only need to consider combinations of d or fewer rows of its generator matrix.

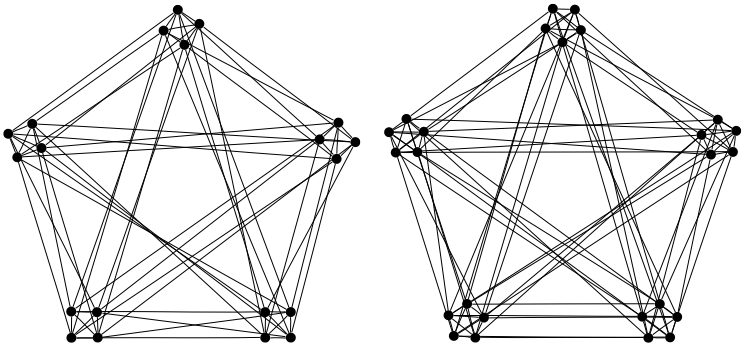
[5] See the web page `http://www.codetables.de/` for details on how codes over $\mathrm{GF}(4)$ of length 18 and 21 can be obtained.

Table 6: *Highest Found Minimum Distance of Codes over* $\mathrm{GF}(m^2)$ *of Length n*

$n \backslash m$	2	3	4	5
2	**2**	**2**	**2**	**2**
3	**2**	**2**	**2**	**2**
4	2	**3***	**3***	**3***
5	**3**	**3**	**3**	**3**
6	**4**	**4**	**4**	**4**
7	3	**4**	**4**	**4**
8	4	4	4	4
9	4	**5**^{s}	**5**	**5**
10	4	**6**	**6**	**6**
11	5^s	5	**6**	**6**
12	6	6	6	6
13	5	6	**7**	**7**
14	6	6	7	**8**
15	6	6	7	7
16	6	6	8	8
17	7	7	8	**9**
18	8*	8	8	**10**
19	7	8		
20	8	8		
21	8*	8		
22	8	9		
23	8	9		
24	8	9		
25	8			
26	8			
27	9^s			
28	10			
29	11			
30	12			



(a) $(6, 2^6, 4)$ **(b)** $(12, 2^{12}, 6)$



(c) $(20, 2^{20}, 8)$ **(d)** $(25, 2^{25}, 8)$

Fig. 4: *Examples of Nested Clique Graphs Corresponding to Codes over* $\mathrm{GF}(4)$

Circulant graphs must be *regular*, i.e., all vertices must have the same number of neighbours. We have previously discovered [27, 28] that many strong circulant self-dual additive codes over $\mathrm{GF}(4)$ can be represented as highly structured *nested clique graphs*. Some of these graphs are shown in Fig. 4. For instance, Fig. 4b shows a graph representation of the $(12, 2^{12}, 6)$ *Dodecacode* consisting of three 4-cliques. The remaining edges form a *Hamiltonian cycle*, i.e., a cycle that visits every vertex of the graph exactly once. Notice that all graphs shown in Fig. 4 have *minimum regular vertex degree*, i.e., each vertex has $d - 1$ neighbours, where d is the distance of the corresponding code.

We have discovered some new highly structured weighted graph representations of self-dual additive codes over $\mathrm{GF}(9)$ and $\mathrm{GF}(16)$. Fig. 5 shows two interconnected 5-cliques where all edges have weight

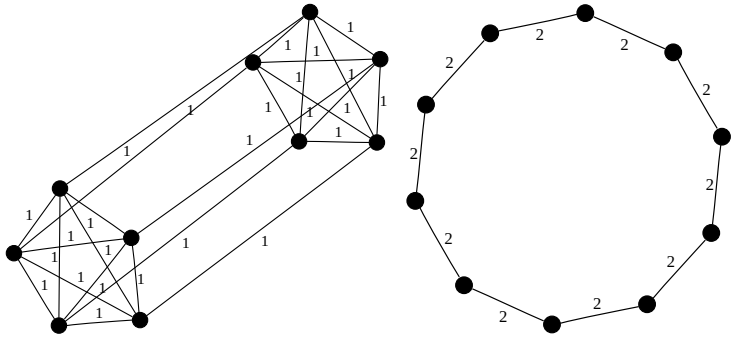


Fig. 5: *Two Graphs Whose Sum Corresponds to the* $(10, 3^{10}, 6)$ *Code*

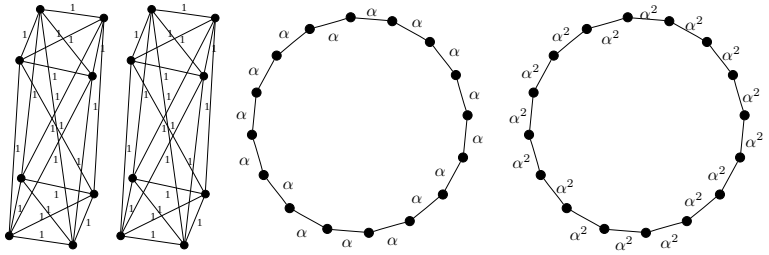


Fig. 6: *Three Graphs Whose Sum Corresponds to the* $(16, 4^{16}, 8)$ *Code*

1, and a 10-cycle where all edges have weight 2. The sum of these two graphs, such that no edges overlap, corresponds to the $(10, 3^{10}, 6)$ code. Up to isomorphism, there is only one way to add a Hamiltonian cycle of weight 2 edges to the double 5-clique, since there cannot be both weight 1 and weight 2 edges between the same pair of vertices. The first row of a circulant generator matrix corresponding to this graph is $(\omega 012111210)$.

As a second example, Fig. 6 shows two pairs of 4-cliques, each of which is connected by a length 8 cycle, and two 16-cycles where all edges have weight α and α^2, respectively, where α is a primitive element of GF(4). The $(16, 4^{16}, 8)$ code generated by $(\omega 0 \alpha^2 1 \alpha 100010001 \alpha 1 \alpha^2)$ corresponds to a sum of these three graphs.

Note that the vertices of the graphs corresponding to the $(10, 3^{10}, 6)$ and $(16, 4^{16}, 8)$ have degree higher than $d - 1$. We have tried to obtain similar graph representations for other codes in Table 6, but without success. Many of the circulant graph codes have vertex de-

gree much higher than $d-1$, for instance the $(14, 5^{14}, 8)$ code generated by $(\omega 1221202021221)$, and the $(18, 5^{18}, 10)$ code generated by $(\omega 12134242124243121)$.

ACKNOWLEDGEMENTS The author would like to thank Matthew G. Parker for helpful discussions and comments. Also thanks to Markus Grassl for helpful comments. This research was supported by the Research Council of Norway.

REFERENCES

[1] CALDERBANK, A. R., RAINS, E. M., SHOR, P. M., SLOANE, N. J. A.: Quantum error correction via codes over GF(4). *IEEE Trans. Inform. Theory* 44(4), 1369–1387, 1998. arXiv:quant-ph/9608006.

[2] RAINS, E. M.: Nonbinary quantum codes. *IEEE Trans. Inform. Theory* 45(6), 1827–1832, 1999. arXiv:quant-ph/9703048.

[3] KETKAR, A., KLAPPENECKER, A., KUMAR, S., SARVEPALLI, P. K.: Nonbinary stabilizer codes over finite fields. *IEEE Trans. Inform. Theory* 52(11), 4892–4914, 2006. arXiv:quant-ph/0508070.

[4] ASHIKHMIN, A., KNILL, E.: Nonbinary quantum stabilizer codes. *IEEE Trans. Inform. Theory* 47(7), 3065–3072, 2001. arXiv:quant-ph/0005008.

[5] SCHLINGEMANN, D.: Stabilizer codes can be realized as graph codes. *Quantum Inf. Comput.* 2(4), 307–323, 2002. arXiv:quant-ph/0111080.

[6] GRASSL, M., KLAPPENECKER, A., RÖTTELER, M.: Graphs, quadratic forms, and quantum codes. In *Proc. IEEE Int. Symp. Inform. Theory*, p. 45. 2002. arXiv:quant-ph/0703112.

[7] GRASSL, M., BETH, T., RÖTTELER, M.: On optimal quantum codes. *Internat. J. Quantum Inf.* 2(1), 55–64, 2004. arXiv:quant-ph/0312164.

[8] HOSTENS, E., DEHAENE, J., DE MOOR, B.: Stabilizer states and Clifford operations for systems of arbitrary dimensions and modular arithmetic. *Phys. Rev. A* 71(4), 2005. arXiv:quant-ph/0408190.

[9] SCHLINGEMANN, D., WERNER, R. F.: Quantum error-correcting codes associated with graphs. *Phys. Rev. A* 65(1), 2002. arXiv:quant-ph/0012111.

[10] BOUCHET, A.: Graphic presentations of isotropic systems. *J. Combin. Theory Ser. B* 45(1), 58–76, 1988.

[11] VAN DEN NEST, M.: *Local Equivalence of Stabilizer States and Codes*. Ph.D. thesis, K. U. Leuven, Belgium, May 2005.

[12] Van den Nest, M., Dehaene, J., De Moor, B.: Graphical description of the action of local Clifford transformations on graph states. *Phys. Rev. A* 69(2), 2004. arXiv:quant-ph/0308151.

[13] Glynn, D. G., Gulliver, T. A., Maks, J. G., Gupta, M. K.: The geometry of additive quantum codes, 2004. Submitted to Springer.

[14] Danielsen, L. E., Parker, M. G.: On the classification of all self-dual additive codes over GF(4) of length up to 12. *J. Combin. Theory Ser. A* 113(7), 1351–1367, 2006. arXiv:math/0504522.

[15] Bahramgiri, M., Beigi, S.: Graph states under the action of local Clifford group in non-binary case, 2006. Preprint. arXiv:quant-ph/0610267.

[16] Klappenecker, A., Rötteler, M.: Beyond stabilizer codes I: Nice error bases. *IEEE Trans. Inform. Theory* 48(8), 2392–2395, 2002. arXiv:quant-ph/0010082.

[17] Gottesman, D.: *Stabilizer Codes and Quantum Error Correction*. Ph.D. thesis, Caltech, May 1997. arXiv:quant-ph/9705052.

[18] Raussendorf, R., Browne, D. E., Briegel, H. J.: Measurement-based quantum computation on cluster states. *Phys. Rev. A* 68(2), 2003. arXiv:quant-ph/0301052.

[19] White, G., Grassl, M.: A new minimum weight algorithm for additive codes. In *Proc. IEEE Int. Symp. Inform. Theory*, pp. 1119–1123. 2006.

[20] Höhn, G.: Self-dual codes over the Kleinian four group. *Math. Ann.* 327(2), 227–255, 2003. arXiv:math/0005266.

[21] Knill, E., Laflamme, R.: Theory of quantum error-correcting codes. *Phys. Rev. A* 55(2), 900–911, 1997. arXiv:quant-ph/9604034.

[22] Rötteler, M., Grassl, M., Beth, T.: On quantum MDS codes. In *Proc. IEEE Int. Symp. Inform. Theory*, p. 355. 2004.

[23] McKay, B. D.: *nauty User's Guide*, 2003. `http://cs.anu.edu.au/~bdm/nauty/`.

[24] Sloane, N. J. A., Plouffe, S.: *The Encyclopedia of Integer Sequences*. Academic Press, San Diego, CA, 1995.

[25] Gaborit, P., Huffman, W. C., Kim, J.-L., Pless, V.: On additive GF(4) codes. In *Codes and Association Schemes*, *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, vol. 56, pp. 135–149. Amer. Math. Soc., Providence, RI, 2001.

[26] Gulliver, T. A., Kim, J.-L.: Circulant based extremal additive self-dual codes over GF(4). *IEEE Trans. Inform. Theory* 50(2), 359–366, 2004.

[27] Danielsen, L. E., Parker, M. G.: Spectral orbits and peak-to-average power ratio of Boolean functions with respect to the $\{I, H, N\}^n$ transform. In *Sequences and Their Applications – SETA 2004*, *Lecture Notes in Comput. Sci.*, vol. 3486, pp. 373–388. Springer, Berlin, 2005. arXiv:cs/0504102.

[28] Danielsen, L. E.: *On Self-Dual Quantum Codes, Graphs, and Boolean Functions*. Master's thesis, Dept. Informat., Univ. Bergen, Norway, Mar. 2005. arXiv:quant-ph/0503236.

PAPER III

EDGE LOCAL COMPLEMENTATION AND EQUIVALENCE OF BINARY LINEAR CODES

Lars Eirik Danielsen Matthew G. Parker

EDGE LOCAL COMPLEMENTATION AND EQUIVALENCE OF BINARY LINEAR CODES

Lars Eirik Danielsen* Matthew G. Parker*

Orbits of graphs under the operation edge local complementation (ELC) are defined. We show that the ELC orbit of a bipartite graph corresponds to the equivalence class of a binary linear code. The information sets and the minimum distance of a code can be derived from the corresponding ELC orbit. By extending earlier results on local complementation (LC) orbits, we classify the ELC orbits of all graphs on up to 12 vertices. We also give a new method for classifying binary linear codes, with running time comparable to the best known algorithm.

1 INTRODUCTION

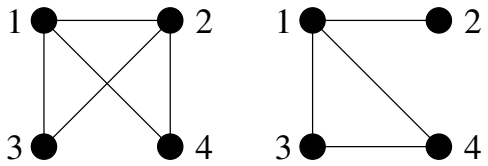
In this section we first give some definitions from graph theory, in particular we describe the two graph operations *local complementation* (LC) and *edge local complementation* (ELC), the latter also known as the *pivot* operation. We then give some definitions related to *binary linear codes*. Of particular interest is the concept of *code equivalence*. Östergård [1] represented codes as graphs, and devised an algorithm for classifying codes up to equivalence. In Section 2, we show a different way of representing a binary linear code as a *bipartite* graph. We prove that ELC on this graph provides a simple way of jumping between equivalent codes, and that the orbit of a bipartite graph under ELC corresponds

*Department of Informatics, University of Bergen, PB 7803, N-5020 Bergen, Norway.

to the complete equivalence class of the corresponding code. We also show how ELC on a bipartite graph generates all *information sets* of the corresponding code. Finally, we show that the *minimum distance* of a code is related to the minimum vertex degree over the corresponding ELC orbit. In Section 3 we describe our algorithm for classifying ELC orbits, which we have used to generate all ELC orbits of graphs on up to 12 vertices. Although ELC orbits of non-bipartite graphs do not have any obvious applications to classical coding theory, they are of interest in other contexts, such as *interlace polynomials* [2, 3] and *quantum graph states* [4] which are related to *quantum error correcting codes*. From the ELC orbits of bipartite graphs a classification of binary linear codes can be derived. Binary linear codes have previously been classified up to length 14 [1, 5]. We have generated the bipartite ELC orbits of graphs on up to 14 vertices, and this classification can be extended to at least 15 vertices [Sang-il Oum, personal communication], showing that our method is comparable to the best known algorithm. However, the main result of this paper is not a classification of codes, but a new way of representing equivalence classes of codes, and a classification of all ELC orbits of length up to 12.

1.1 Graph Theory

A *graph* is a pair $G = (V, E)$ where V is a set of *vertices*, and $E \subseteq V \times V$ is a set of *edges*. A graph with n vertices can be represented by an $n \times n$ *adjacency matrix* Γ, where $\gamma_{ij} = 1$ if $\{i, j\} \in E$, and $\gamma_{ij} = 0$ otherwise. We will only consider *simple undirected* graphs whose adjacency matrices are symmetric with all diagonal elements being 0, i.e., all edges are bidirectional and no vertex can be adjacent to itself. The *neighbourhood* of $v \in V$, denoted $N_v \subset V$, is the set of vertices connected to v by an edge. The number of vertices adjacent to v is called the *degree* of v. The *induced subgraph* of G on $W \subseteq V$ contains vertices W and all edges from E whose endpoints are both in W. The *complement* of G is found by replacing E with $V \times V - E$, i.e., the edges in E are changed to non-edges, and the non-edges to edges. Two graphs $G = (V, E)$ and $G' = (V, E')$ are *isomorphic* if and only if there exists a permutation π on V such that $\{u, v\} \in E$ if and only if $\{\pi(u), \pi(v)\} \in E'$. A *path* is a sequence of vertices, $(v_1, v_2, \ldots, v_i)$, such that $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{i-1}, v_i\} \in E$. A graph is *connected* if there is a path from any vertex to any other vertex in the graph. A graph is *bipartite* if its set of vertices can be decomposed into two disjoint sets such that no two



(a) *The Graph G* **(b)** *The Graph* $G * 1$

Fig. 1: *Example of Local Complementation*

vertices within the same set are adjacent. We call a graph (a, b)-*bipartite* if its vertices can be decomposed into sets of size a and b.

Definition 1 ([6–8])**.** Given a graph $G = (V, E)$ and a vertex $v \in V$, let $N_v \subset V$ be the neighbourhood of v. *Local complementation* (LC) on v transforms G into $G * v$ by replacing the induced subgraph of G on N_v by its complement. (Fig. 1)

Definition 2 ([7])**.** Given a graph $G = (V, E)$ and an edge $\{u, v\} \in E$, *edge local complementation* (ELC) on $\{u, v\}$ transforms G into $G^{(uv)} = G * u * v * u = G * v * u * v$.

Definition 3 ([7])**.** ELC on $\{u, v\}$ can equivalently be defined as follows. Decompose $V \setminus \{u, v\}$ into the following four disjoint sets, as visualized in Fig. 2.

- A Vertices adjacent to u, but not to v.
- B Vertices adjacent to v, but not to u.
- C Vertices adjacent to both u and v.
- D Vertices adjacent to neither u nor v.

To obtain $G^{(uv)}$, perform the following procedure. For any pair of vertices $\{x, y\}$, where x belongs to class A, B, or C, and y belongs to a different class A, B, or C, "toggle" the pair $\{x, y\}$, i.e., if $\{x, y\} \in E$, delete the edge, and if $\{x, y\} \notin E$, add the edge $\{x, y\}$ to E. Finally, swap the labels of vertices u and v.

Definition 4. The *LC orbit* of a graph G is the set of all graphs that can be obtained by performing any sequence of LC operations on G.

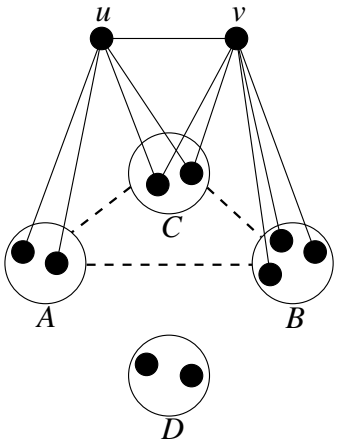


Fig. 2: *Visualization of the ELC Operation*

Similarly, the *ELC orbit* of G comprises all graphs that can be obtained by performing any sequence of ELC operations on G. (Usually we consider LC and ELC orbits of unlabeled graphs. In the cases where we consider orbits of labeled graphs, this will be noted.)

The LC operation was first defined by de Fraysseix [8], and later studied by Fon-der-Flaas [6] and Bouchet [7]. Bouchet defined ELC as "complementation along an edge" [7], but this operation is also known as *pivoting* on a graph [2, 9]. LC orbits of graphs have been used to study *quantum graph states* [10–12], which are equivalent to *self-dual additive codes over* $\mathrm{GF}(4)$ [13]. We have previously used LC orbits to classify such codes [14, 15]. ELC orbits have also been studied in the context of quantum graph states [4, 9]. *Interlace polynomials* of graphs have been defined with respect to both ELC [2] and LC [3]. These polynomials encode properties of the graph orbits, and were originally used to study a problem related to DNA sequencing [16].

Proposition 5. *If $G = (V, E)$ is a connected graph, then, for any vertex $v \in V$, $G * v$ must also be connected. Likewise, for any edge $\{u, v\} \in E$, $G^{(uv)}$ must be connected.*

Proof. If the edge $\{x, y\}$ is deleted as part of an LC operation on v, both x and y must be, and will remain, connected to v. Similarly, if by performing ELC on the edge $\{u, v\}$, the edge $\{x, y\}$ is deleted, both x

and y will remain connected to either u, v, or both, and u and v will remain connected. □

Proposition 6 ([9]). *If G is an (a,b)-bipartite graph, then, for any edge $\{u,v\} \in E$, $G^{(uv)}$ must also be (a,b)-bipartite.*

Proof. A bipartite graph with an edge $\{u,v\}$ can not contain any vertex that is connected to both u and v. Using the terminology of Definition 3, the set C will always be empty when we perform ELC on a bipartite graph. Moreover, all vertices in the set A must belong to the same partition as u, and all vertices in B must belong to the same partition as v. All edges that are added or deleted have one endpoint in A and one in B, and it follows that bipartiteness is preserved. □

Proposition 7. *Let G be a bipartite graph, and let $\{u,v\} \in E$. Then $G^{(uv)}$ can be obtained by "toggling" all edges between the sets $N_u \setminus \{v\}$ and $N_v \setminus \{u\}$, followed by a swapping of vertices u and v.*

1.2 Coding Theory

A binary linear code, $\mathcal{C}$, is a linear subspace of $\mathrm{GF}(2)^n$ of dimension k, where $0 \leq k \leq n$. $\mathcal{C}$ is called an $[n,k]$ code, and the 2^k elements of $\mathcal{C}$ are called *codewords*. The *Hamming weight* of $\boldsymbol{u} \in \mathrm{GF}(2)^n$, denoted $\mathrm{wt}(\boldsymbol{u})$, is the number of nonzero components of $\boldsymbol{u}$. The *Hamming distance* between $\boldsymbol{u}, \boldsymbol{v} \in \mathrm{GF}(2)^n$ is $\mathrm{wt}(\boldsymbol{u} - \boldsymbol{v})$. The *minimum distance* of the code $\mathcal{C}$ is the minimal Hamming distance between any two codewords of $\mathcal{C}$. Since $\mathcal{C}$ is a linear code, the minimum distance is also given by the smallest weight of any codeword in $\mathcal{C}$. A code with minimum distance d is called an $[n,k,d]$ code. A code is *decomposable* if it can be written as the *direct sum* of two smaller codes. For example, let $\mathcal{C}$ be an $[n,k,d]$ code and $\mathcal{C}'$ an $[n',k',d']$ code. The direct sum, $\mathcal{C} \oplus \mathcal{C}' = \{u||v \mid u \in \mathcal{C}, v \in \mathcal{C}'\}$, where $||$ means concatenation, is an $[n+n', k+k', \min\{d,d'\}]$ code. Two codes, $\mathcal{C}$ and $\mathcal{C}'$, are considered to be *equivalent* if one can be obtained from the other by some permutation of the coordinates, or equivalently, a permutation of the columns of a generator matrix. We define the *dual* of the code $\mathcal{C}$ with respect to the standard inner product, $\mathcal{C}^\perp = \{\boldsymbol{u} \in \mathrm{GF}(2)^n \mid \boldsymbol{u} \cdot \boldsymbol{c} = 0 \text{ for all } \boldsymbol{c} \in \mathcal{C}\}$. $\mathcal{C}$ is called *self-dual* if $\mathcal{C} = \mathcal{C}^\perp$, and *isodual* if $\mathcal{C}$ is equivalent to $\mathcal{C}^\perp$. Self-dual and isodual codes must have even length n, and dimension $k = \frac{n}{2}$. The code $\mathcal{C}$ can be defined by a $k \times n$ *generator matrix*, C, whose rows span $\mathcal{C}$. A set of k linearly independent columns of C is called an *information set* of $\mathcal{C}$.

We can permute the columns of C such that an information set makes up the first k columns. By elementary row operations, this matrix can then be transformed into a matrix of the form $C' = (I \mid P)$, where I is a $k \times k$ identity matrix, and P is some $k \times (n-k)$ matrix. The matrix C', which is said to be of *standard form*, generates a code $\mathcal{C}'$ which is equivalent to $\mathcal{C}$. Every code is equivalent to a code with a generator matrix of standard form. The matrix $H' = (P^{\mathrm{T}} \mid I)$, where I is an $(n-k) \times (n-k)$ identity matrix is called the *parity check matrix* of $\mathcal{C}'$. Observe that $G'{H'}^{\mathrm{T}} = \mathbf{0}$, where $\mathbf{0}$ is the all-zero vector. It follows that H' must be the generator matrix of $\mathcal{C}'^{\perp}$.

2 ELC AND CODE EQUIVALENCE

As mentioned earlier, LC orbits of graphs correspond to equivalence classes of self-dual quantum codes. We have previously classified all such codes of length up to 12 [15], by classifying LC orbits of simple undirected graphs. In this paper, we show that ELC orbits of bipartite graphs correspond to the equivalence classes of binary linear codes. First we explain how a binary linear code can be represented by a graph.

Definition 8 ([17, 18])**.** Let $\mathcal{C}$ be a binary linear $[n,k]$ code with generator matrix $C = (I \mid P)$. Then the code $\mathcal{C}$ corresponds to the $(k, n-k)$-bipartite graph on n vertices with adjacency matrix

$$\Gamma = \begin{pmatrix} \mathbf{0}_{k\times k} & P \\ P^{\mathrm{T}} & \mathbf{0}_{(n-k)\times(n-k)} \end{pmatrix},$$

where $\mathbf{0}$ denote all-zero matrices of the specified dimensions.

Theorem 9. *Let $G = (V, E)$ be the $(k, n-k)$-bipartite graph derived from a standard form generator matrix $C = (I \mid P)$ of the $[n,k]$ code $\mathcal{C}$. Let G' be the graph obtained by performing ELC on the edge $\{u, v\} \in E$, followed by a swapping of vertices u and v. Then the code $\mathcal{C}'$ generated by $C' = (I \mid P')$ corresponding to G' is equivalent to $\mathcal{C}$, and can be obtained by interchanging coordinates u and v of $\mathcal{C}$.*

Proof. Assume, without loss of generality, that $u \leq k$ and $v > k$. C' can be obtained from C by adding row u to all rows in $N_v \setminus \{u\}$ and then swapping columns u and v, where N_v denotes the neighbourhood of v in G. These operations preserve the equivalence of linear codes.

As described in Proposition 7, the bipartite graph G is transformed into G' by "toggling" all pairs of vertices $\{x, y\}$, where $x \in N_u \setminus \{v\}$ and $y \in N_v \setminus \{u\}$. This action on the submatrix P is implemented by the row additions on C described above. However, this also "toggles" the pairs $\{v, y\}$, where $y \in N_v \setminus \{u\}$, transforming column v of C into a vector with 0 in all coordinates except u. But column u of C now contains the original column v, and thus swapping columns u and v restores the neighbourhood of v, giving the desired submatrix P. □

Corollary 10. *Applying any sequence of ELC operations to a graph G corresponding to a code $\mathcal{C}$ will produce a graph corresponding to a code equivalent to $\mathcal{C}$.*

Instead of mapping the generator matrix $C = (I \mid P)$ to the adjacency matrix of a bipartite graph in order to perform ELC on the edge $\{u, v\}$, we can work directly with the submatrix P. Let the rows of P be labeled $1, 2, \ldots, k$ and the columns of P be labeled $k+1, k+2, \ldots, n$. Assume that u indicates a row of P and that v indicates a column of P. The element P_{ij} is then replaced by $1 - P_{ij}$ if $i \neq u$, $j \neq v$, and $P_{uj} = P_{iv} = 1$.

Example 11. The $[7, 4, 3]$ Hamming code has a generator matrix

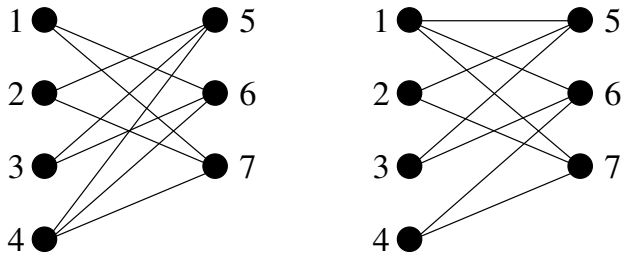
$$C = \left(\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right),$$

which corresponds to the graph shown in Fig. 3a. ELC on the edge $\{2, 7\}$ produces the graph shown in Fig. 3b, which corresponds to the generator matrix

$$C' = \left(\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right).$$

The code generated by C' is also obtained by swapping coordinates 2 and 7 of the code generated by C.

Consider a code $\mathcal{C}$. As described in Section 1.2, it is possible to go from a generator matrix of standard form, $C = (I \mid P)$, to another generator matrix of standard form, C', of a code equivalent to $\mathcal{C}$ by one of the $n!$ possible permutations of the columns of C, followed by



(a) *The Graph G* **(b)** *The Graph* $G^{(27)}$

Fig. 3: *Two Graph Representations of the* $[7, 4, 3]$ *Hamming Code*

elementary row operations. More precisely, we can get from C to C' via a combination of the following operations.

1. Permuting the columns of P.
2. Permuting the columns of I, followed by the same permutation on the rows of C, to restore standard form.
3. Swapping columns from I with columns from P, such that the first k columns still is an information set, followed by some elementary row operations to restore standard form.

Theorem 12. *Let* $\mathcal{C}$ *and* $\mathcal{C}'$ *be equivalent codes. Let* C *and* C' *be matrices of standard form generating* $\mathcal{C}$ *and* $\mathcal{C}'$*. Let* G *and* G' *be the bipartite graphs corresponding to* C *and* C'*.* G' *is isomorphic to a graph obtained by performing some sequence of ELC operations on* G*.*

Proof. $\mathcal{C}$ and $\mathcal{C}'$ must be related by a combination of the operations 1, 2, and 3 listed above. It is easy to see that operations 1 and 2 applied to G produce an isomorphic graph. It remains to prove that operation 3 always corresponds to some sequence of ELC operations. We know from Theorem 9 that swapping columns u and v of C, where u is part of I and v is part of P, corresponds to ELC on the edge $\{u, v\}$ of G, followed by a swapping of the vertices u and v. When $\{u, v\}$ is not an edge of G, we can not swap columns u and v of C via ELC. In this case, coordinate v of column u is 0, and column u has 1 in coordinate u and 0 elsewhere. Swapping these columns would result in a generator matrix where the first k columns all have 0 at coordinate u. These columns can not correspond to an information set. It follows that if $\{u, v\}$ is not an

edge of G, swapping columns u and v is not a valid operation of type 3 in the above list. Thus ELC and graph isomorphism cover all possible operations that map standard form generator matrices of equivalent codes to each other. □

Let us for a moment consider ELC orbits of *labeled* graphs, i.e., where we do not take isomorphism into consideration. Let $G = (V, E)$ be the connected bipartite graph representing the indecomposable code $\mathcal{C}$, and $G^{(uv)}$ be the graph obtained by ELC on the edge $\{u, v\} \in E$. Since we perform ELC on $\{u, v\}$ without swapping u and v afterwards, the adjacency matrix of $G^{(uv)}$ will not be of the type we saw in Definition 8. Assuming that vertices $\{1, 2, \ldots, k\}$ make up one of the partitions of the bipartite graph G, we can think of G as a graph corresponding to the information set $\{1, 2, \ldots, k\}$ of $\mathcal{C}$. Assume that $u \leq k$ and $v > k$. $G^{(uv)}$ will then represent another information set of $\mathcal{C}$, namely $\{1, 2, \ldots, k\} \setminus \{u\} \cup \{v\}$.

Theorem 13. *Let G be a connected bipartite graph representing the indecomposable code $\mathcal{C}$. Each labeled graph in the ELC orbit of G corresponds to an information set of $\mathcal{C}$. If $\mathcal{C}$ is a self-dual code, each graph corresponds to two information sets, one for each partition. Moreover, the number of information sets of $\mathcal{C}$ equals the number of labeled graphs in the ELC orbit of G, or twice the number of graphs if $\mathcal{C}$ is a self-dual code.*

Proof. Performing ELC without swapping vertices afterwards corresponds to elementary row operations on the associated generator matrix, and will thus leave the code invariant. The only thing we change with ELC is the information set of the code, as indicated by the bipartition of the graph. We know from Theorem 12 that if two generator matrices of standard form generate equivalent codes, we can always get from one to the other via ELC operations on the associated graph. It follows from this that when we consider labeled graphs, and do not swap vertices to obtain a code of standard form, we find all information sets in the ELC orbit. If and only if a code is self-dual, $(I \mid P)$ will generate the same code as $(P^{\mathrm{T}} \mid I)$. Since the matrices $(I \mid P)$ and $(P^T \mid I)$ correspond to exactly the same graph, but two different information sets, we must multiply the ELC orbit size with two to get the number of information sets of a self-dual code. □

Note that the distinction between ELC with or without a final swapping of vertices is only significant when we want to find information

Table 1: *Numbers of LC Orbits*

n	1	2	3	4	5	6	7	8	9	10	11	12
i_n^{LC}	1	1	1	2	4	11	26	101	440	3132	40 457	1 274 068
t_n^{LC}	1	2	3	6	11	26	59	182	675	3990	45 144	1 323 363

sets. For other applications, where we consider graphs up to isomorphism, this distinction is not of importance.

Theorem 14. *The minimum distance, d, of a binary linear $[n, k, d]$ code $\mathcal{C}$, is equal to $\delta + 1$, where δ is the smallest vertex degree over all graphs in the associated ELC orbit.*

Proof. A vertex with degree $d - 1$ in the ELC orbit corresponds to a codeword of weight d. We need to show that such a vertex always exists. Let C be a generator matrix of standard form, where all rows have weight greater than d, that generates a code equivalent to $\mathcal{C}$. Find a codeword c of weight d, generated by C, and let the i-th row of C be one of the rows that c is linearly dependent on. Permute the columns of C to obtain C' where the first k columns is still an information set, and where c is mapped to c' with 1 in coordinate i, with the rest of the k first coordinates being 0. (This will always be possible, since the i-th row of C has weight greater than d.) Replace the i-th row of C' by c' to get C''. We can transform C'' into a matrix of the form $(I \mid P)$ by elementary row operations. Row i of this final matrix has weight d, and thus the corresponding bipartite graph has a vertex with degree $d - 1$. □

3 Classification of ELC Orbits

We have previously classified all self-dual additive codes over GF(4) of length up to 12 [15, 19], by classifying orbits of simple undirected graphs with respect to local complementation and graph isomorphism. In Table 1, the sequence (i_n^{LC}) gives the number of LC orbits of connected graphs on n vertices, while (t_n^{LC}) gives the total number of LC orbits of graphs on n vertices. A database containing one representative from each LC orbit is available at `http://www.ii.uib.no/~larsed/vncorbits/`.

By recursively applying ELC operations to all edges of a graph, whilst checking for graph isomorphism using the program *nauty* [20],

we can find all members of the ELC orbit. Let $\boldsymbol{G}_n$ be the set of all unlabeled simple undirected connected graphs on n vertices. Let the set of all distinct ELC orbits of connected graphs on n vertices be a partitioning of $\boldsymbol{G}_n$ into i_n^{ELC} disjoint sets. Our previous classification of the LC orbits of all graphs of up to 12 vertices helps us to classify ELC orbits, since it follows from Definition 2 that each LC orbit can be partitioned into a set of disjoint ELC orbits. We have used this fact to classify all ELC orbits of graphs on up to 12 vertices, a computation that required approximately one month of running time on a parallel cluster computer. In Table 2, the sequence (i_n^{ELC}) gives the number of ELC orbits of connected graphs on n vertices, while (t_n^{ELC}) gives the total number of ELC orbits of graphs on n vertices. Note that the value of t_n can be derived easily once the sequence (i_m) is known for $1 \leq m \leq n$, using the *Euler transform* [21],

$$\begin{aligned} c_n &= \sum_{d|n} d i_d, \\ t_1 &= c_1, \\ t_n &= \frac{1}{n}\left(c_n + \sum_{k=1}^{n-1} c_k t_{n-k}\right). \end{aligned}$$

A database containing one representative from each ELC orbit can be found at `http://www.ii.uib.no/~larsed/pivot/`.

We are particularly interested in bipartite graphs, because of their connection to binary linear codes. For the classification of the orbits of bipartite graphs with respect to ELC and graph isomorphism, the following technique is helpful. If G is an (a, b)-bipartite graph, it has $2^a + 2^b - 2$ possible *extensions*. Each extension is formed by adding a new vertex and joining it to all possible combinations of at least one of the old vertices. Let $\boldsymbol{P}_n$ be a set containing one representative from each ELC orbit of all connected bipartite graphs on n vertices. The set $\boldsymbol{E}_n$ is formed by making all possible extensions of all graphs in $\boldsymbol{P}_{n-1}$. It can then be shown that $\boldsymbol{P}_n \subset \boldsymbol{E}_n$, i.e., that the set $\boldsymbol{E}_n$ will contain at least one representative from each ELC orbit of connected bipartite graphs on n vertices. The set $\boldsymbol{E}_n$ will be much smaller than $\boldsymbol{G}_n$, so it will be more efficient to search for a set of ELC orbit representatives within $\boldsymbol{E}_n$. A similar technique was used by Glynn et al. [10] to classify LC orbits.

In Table 2, the sequence $(i_n^{ELC,B})$ gives the number of ELC orbits of connected bipartite graphs on n vertices, and $(t_n^{ELC,B})$ gives the total

Table 2: *Numbers of ELC Orbits and Binary Linear Codes*

n	i_n^{ELC}	t_n^{ELC}	$i_n^{ELC,B}$	$t_n^{ELC,B}$	i_n^{C}	$i_n^{C_{iso}}$
1	1	1	1	1	1	-
2	1	2	1	2	1	1
3	2	4	1	3	2	-
4	4	9	2	6	3	1
5	10	21	3	10	6	-
6	35	64	8	22	13	3
7	134	218	15	43	30	-
8	777	1068	43	104	76	10
9	6702	8038	110	250	220	-
10	104 825	114 188	370	720	700	40
11	3 370 317	3 493 965	1260	2229	2520	-
12	231 557 290	235 176 097	5366	8361	10 503	229
13	*?*	*?*	25 684	36 441	51 368	-
14			154 104	199 610	306 328	1880
15			1 156 716	1 395 326	2 313 432	-
16			*?*	*?*	*23 069 977*	*?*
17			*157 302 628*	*?*	*314 605 256*	-

number of ELC orbits of bipartite graphs on n vertices. A database containing one representative from each of these orbits can be found at `http://www.ii.uib.no/~larsed/pivot/`.

Theorem 15. *Let $k \neq \frac{n}{2}$. Then the number of inequivalent binary linear $[n,k]$ codes, which is also the number of inequivalent $[n,n-k]$ codes, is equal to the number of ELC orbits of $(n-k,k)$-bipartite graphs.*

When n is even and $k = \frac{n}{2}$, the number of inequivalent binary linear $[n,k]$ codes is equal to twice the number of ELC orbits of (k,k)-bipartite graphs minus the number of isodual codes of length n.

Proof. We recall that if a code $\mathcal{C}$ is generated by $(I \mid P)$, then its dual, $\mathcal{C}^{\perp}$, is generated by $(P^{\mathrm{T}} \mid I)$. Also note that $\mathcal{C}^{\perp}$ is equivalent to the code generated by $(I \mid P^{\mathrm{T}})$. The bipartite graphs corresponding to the codes generated by $(I \mid P)$ and $(I \mid P^{\mathrm{T}})$ are isomorphic. It follows that the ELC orbit associated with an $[n,k]$ code $\mathcal{C}$ is simultaneously the orbit associated with the dual $[n,n-k]$ code $\mathcal{C}^{\perp}$. In the case where $k = \frac{n}{2}$, each ELC orbit corresponds to two non-equivalent $[n,k]$ codes, except in the case where $\mathcal{C}$ is isodual. □

Corollary 16. *The total number of binary linear codes of length n is equal to twice the number of ELC orbits of bipartite graphs on n vertices, minus the number of isodual codes of length n.*

Note that if we only consider connected graphs on n vertices, we get the number of indecomposable codes of length n, i_n^C, i.e., the codes that can not be written as the direct sum of two smaller codes. The total number of codes can easily be derived from the values of (i_n^C). Table 2 gives the number of ELC orbits of connected bipartite graphs on n vertices, $i_n^{ELC,B}$, the number of indecomposable binary linear codes of length n, i_n^C, and the number of indecomposable isodual codes of length n, $i_n^{C_{iso}}$. A method for counting the number of binary linear codes by using computer algebra tools was devised by Fripertinger and Kerber [22]. A table enumerating binary linear codes of length up to 25 is available online at `http://www.mathe2.uni-bayreuth.de/frib/codes/tables_2.html`. The numbers in italics in Table 2 are taken from this web page. Note however that this approach only gives the number of inequivalent codes, and does not produce the codes themselves. Classification of all binary linear codes of length up to 14 and with distance at least 3 was carried out by Östergård [1]. He also used a graph-based algorithm, but one quite different from the method described in this paper. In a recent book by Kaski and Östergård [5], it is proposed as a research problem to extend this classification to lengths higher than 14. Sang-il Oum [personal communication] demonstrated that the 1 395 326 ELC orbits of bipartite graphs on 15 vertices can be generated in about 58 hours. This indicates that classification of codes by ELC orbits is comparable to the currently best known algorithm. It may also be possible that our method will be more efficient than existing algorithms for classifying special types of codes. For instance, matrices of the form $(I \mid P)$, where P is symmetric, generate a subset of the isodual codes. The bipartite graphs corresponding to these codes, which were also studied by Curtis [17], should be well suited to our method, since any graph of this type must arise as an extension of a graph of the same type.

ACKNOWLEDGEMENTS This research was supported by the Research Council of Norway. We would like to thank the Bergen Center for Computational Science, whose cluster computer made the results in this paper possible. Thanks to Joakim G. Knudsen for help with improving Theorem 13.

REFERENCES

[1] ÖSTERGÅRD, P. R. J.: Classifying subspaces of Hamming spaces. *Des. Codes Cryptogr.* 27(3), 297–305, 2002.

[2] ARRATIA, R., BOLLOBÁS, B., SORKIN, G. B.: The interlace polynomial of a graph. *J. Combin. Theory Ser. B* 92(2), 199–233, 2004. arXiv:math.CO/0209045.

[3] AIGNER, M., VAN DER HOLST, H.: Interlace polynomials. *Linear Algebra Appl.* 377, 11–30, 2004.

[4] VAN DEN NEST, M., DE MOOR, B.: Edge-local equivalence of graphs, 2005. Preprint. arXiv:math.CO/0510246.

[5] KASKI, P., ÖSTERGÅRD, P. R. J.: *Classification Algorithms for Codes and Designs*, *Algorithms and Computation in Mathematics*, vol. 15. Springer, Berlin, 2006.

[6] FON-DER FLAAS, D. G.: On local complementations of graphs. In *Combinatorics (Eger, 1987)*, *Colloq. Math. Soc. János Bolyai*, vol. 52, pp. 257–266. North-Holland, Amsterdam, 1988.

[7] BOUCHET, A.: Graphic presentations of isotropic systems. *J. Combin. Theory Ser. B* 45(1), 58–76, 1988.

[8] DE FRAYSSEIX, H.: Local complementation and interlacement graphs. *Discrete Math.* 33(1), 29–35, 1981.

[9] RIERA, C., PARKER, M. G.: On pivot orbits of Boolean functions. In *Fourth International Workshop on Optimal Codes and Related Topics*, pp. 248–253. Bulgarian Acad. Sci. Inst. Math. Inform., Sofia, 2005.

[10] GLYNN, D. G., GULLIVER, T. A., MAKS, J. G., GUPTA, M. K.: The geometry of additive quantum codes, 2004. Submitted to Springer.

[11] HEIN, M., EISERT, J., BRIEGEL, H. J.: Multi-party entanglement in graph states. *Phys. Rev. A* 69(6), 2004. arXiv:quant-ph/0307130.

[12] VAN DEN NEST, M., DEHAENE, J., DE MOOR, B.: Graphical description of the action of local Clifford transformations on graph states. *Phys. Rev. A* 69(2), 2004. arXiv:quant-ph/0308151.

[13] CALDERBANK, A. R., RAINS, E. M., SHOR, P. M., SLOANE, N. J. A.: Quantum error correction via codes over GF(4). *IEEE Trans. Inform. Theory* 44(4), 1369–1387, 1998. arXiv:quant-ph/9608006.

[14] DANIELSEN, L. E., PARKER, M. G.: Spectral orbits and peak-to-average power ratio of Boolean functions with respect to the $\{I, H, N\}^n$ transform. In *Sequences and Their Applications – SETA 2004*, *Lecture Notes in Comput. Sci.*, vol. 3486, pp. 373–388. Springer, Berlin, 2005. arXiv:cs.IT/0504102.

[15] Danielsen, L. E., Parker, M. G.: On the classification of all self-dual additive codes over GF(4) of length up to 12. *J. Combin. Theory Ser. A* 113(7), 1351–1367, 2006. arXiv:math.CO/0504522.

[16] Arratia, R., Bollobás, B., Coppersmith, D., Sorkin, G. B.: Euler circuits and DNA sequencing by hybridization. *Discrete Appl. Math.* 104(1–3), 63–96, 2000.

[17] Curtis, R. T.: On graphs and codes. *Geom. Dedicata* 41(2), 127–134, 1992.

[18] Parker, M. G., Rijmen, V.: The quantum entanglement of binary and bipolar sequences. In *Sequences and Their Applications – SETA 2001*, Discrete Math. Theor. Comput. Sci., pp. 296–309. Springer, London, 2002. arXiv:quant-ph/0107106.

[19] Danielsen, L. E.: *On Self-Dual Quantum Codes, Graphs, and Boolean Functions*. Master's thesis, Dept. Informat., Univ. Bergen, Norway, Mar. 2005. arXiv:quant-ph/0503236.

[20] McKay, B. D.: *nauty User's Guide, Version 2.2*, 2003. `http://cs.anu.edu.au/~bdm/nauty/`.

[21] Sloane, N. J. A., Plouffe, S.: *The Encyclopedia of Integer Sequences*. Academic Press, San Diego, CA, 1995.

[22] Fripertinger, H., Kerber, A.: Isometry classes of indecomposable linear codes. In *Applied algebra, algebraic algorithms and error-correcting codes*, *Lecture Notes in Comput. Sci.*, vol. 948, pp. 194–204. Springer, Berlin, 1995.

PAPER IV

SPECTRAL ORBITS AND PEAK-TO-AVERAGE POWER RATIO OF BOOLEAN FUNCTIONS WITH RESPECT TO THE $\{I, H, N\}^n$ TRANSFORM

Lars Eirik Danielsen Matthew G. Parker

SPECTRAL ORBITS AND PEAK-TO-AVERAGE POWER RATIO OF BOOLEAN FUNCTIONS WITH RESPECT TO THE $\{I, H, N\}^n$ TRANSFORM

Lars Eirik Danielsen* Matthew G. Parker*

We enumerate the inequivalent self-dual additive codes over GF(4) of length up to 12. These codes have a well-known interpretation as quantum codes. They can also be represented by graphs, where a simple graph operation generates the orbits of equivalent codes. We highlight the regularity and structure of some graphs that correspond to codes with high minimum distance. The codes can also be interpreted as quadratic Boolean functions, where inequivalence takes on a spectral meaning. In this context we define PAR_{IHN}, peak-to-average power ratio with respect to the $\{I, H, N\}^n$ transform set. We prove that PAR_{IHN} of a quadratic Boolean function is equivalent to the the size of the maximum independent set over the associated orbit of graphs. Finally we propose a construction technique to generate nonquadratic Boolean functions with low PAR_{IHN}.

1 Self-Dual Additive Codes over GF(4)

A quantum error-correcting code with parameters $[[n, k, d]]$ encodes k qubits in an entangled state of n qubits such that any error affecting

*Department of Informatics, University of Bergen, PB 7803, N-5020 Bergen, Norway.

Table 1: *Number of Inequivalent Indecomposable (i_n) and (Possibly) Decomposable (t_n) Self-Dual Additive Codes over* GF(4)

n	1	2	3	4	5	6	7	8	9	10	11	12
i_n	1	1	1	2	4	11	26	101	440	3132	40 457	1 274 068
t_n	1	2	3	6	11	26	59	182	675	3990	45 144	1 323 363

less than d qubits can be detected, and any error affecting at most $\left\lfloor \frac{d-1}{2} \right\rfloor$ qubits can be corrected. A quantum code of the stabilizer type corresponds to a code $\mathcal{C} \subset \mathrm{GF}(4)^n$ [1]. We denote $\mathrm{GF}(4) = \{0, 1, \omega, \omega^2\}$, where $\omega^2 = \omega + 1$. *Conjugation* in $\mathrm{GF}(4)$ is defined by $\overline{x} = x^2$. The *trace map*, $\mathrm{Tr} : \mathrm{GF}(4) \to \mathrm{GF}(2)$, is defined by $\mathrm{Tr}(x) = x + \overline{x}$. The *trace inner product* of two vectors of length n over $\mathrm{GF}(4)$, $\boldsymbol{u}$ and $\boldsymbol{v}$, is given by $\boldsymbol{u} * \boldsymbol{v} = \sum_{i=1}^{n} \mathrm{Tr}(u_i \overline{v_i})$. Because of the structure of stabilizer codes, the corresponding code over $\mathrm{GF}(4)$, $\mathcal{C}$, will be *additive* and satisfy $\boldsymbol{u} * \boldsymbol{v} = 0$ for any two codewords $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{C}$. This is equivalent to saying that the code must be *self-orthogonal* with respect to the trace inner product, i.e., $\mathcal{C} \subseteq \mathcal{C}^{\perp}$, where $\mathcal{C}^{\perp} = \{\boldsymbol{u} \in \mathrm{GF}(4)^n \mid \boldsymbol{u} * \boldsymbol{c} = 0, \forall \boldsymbol{c} \in \mathcal{C}\}$.

We will only consider codes of the special case where the dimension $k = 0$. Zero-dimensional quantum codes can be understood as highly entangled single quantum states which are robust to error. These codes map to additive codes over $\mathrm{GF}(4)$ which are *self-dual* [2], $\mathcal{C} = \mathcal{C}^{\perp}$. The number of inequivalent self-dual additive codes over $\mathrm{GF}(4)$ of length n has been classified by Calderbank et al. [1] for $n \leq 5$, by Höhn [3] for $n \leq 7$, by Hein et al. [4] for $n \leq 7$, and by Glynn et al. [5] for $n \leq 9$. Moreover, Glynn has recently posted these results as sequence A090899 in *The On-Line Encyclopedia of Integer Sequences* [6]. We extend this sequence from $n = 9$ to $n = 12$ both for indecomposable and decomposable codes as shown in Table 1. Table 2 shows the number of inequivalent indecomposable codes by minimum distance. The minimum distance, d, of a self-dual additive code over $\mathrm{GF}(4)$, $\mathcal{C}$, is the smallest weight (i.e., number of nonzero components) of any nonzero codeword in $\mathcal{C}$. A database of orbit representatives with information about orbit size, minimum distance, and weight distribution is also available at `http://www.ii.uib.no/~larsed/vncorbits/`.

Table 2: *Number of Indecomposable Self-Dual Additive Codes over* GF(4) *by Distance*

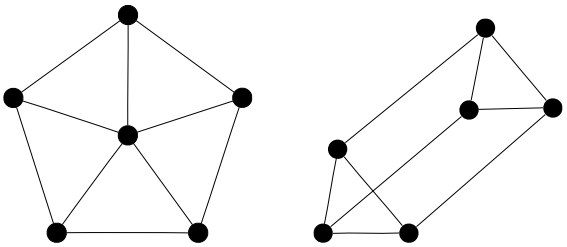
$d\backslash n$	2	3	4	5	6	7	8	9	10	11	12
2	1	1	2	3	9	22	85	363	2436	26 750	611 036
3				1	1	4	11	69	576	11 200	467 513
4					1		5	8	120	2506	195 455
5										1	63
6											1
All	1	1	2	4	11	26	101	440	3132	40 457	1 274 068

2 Graphs, Boolean Functions, and LC-Equivalence

A self-dual additive code over GF(4) corresponds to a *graph state* [4] if its generator matrix, G, can be written as $G = \Gamma + \omega I$, where Γ is a symmetric matrix over GF(2) with zeros on the diagonal. The matrix Γ can be interpreted as the adjacency matrix of a simple undirected graph on n vertices. It has been shown by Schlingemann and Werner [7], Grassl et al. [8], Glynn [9], and Van den Nest et al. [10] that all stabilizer states can be transformed into an equivalent graph state. Thus all self-dual additive codes over GF(4) can be represented by graphs. These codes also have another interpretation as quadratic Boolean functions of n variables. A quadratic function, f, can be represented by an adjacency matrix, Γ, where $\Gamma_{i,j} = \Gamma_{j,i} = 1$ if $x_i x_j$ occurs in f, and $\Gamma_{i,j} = 0$ otherwise.

Example 1. A self-dual additive code over GF(4) with parameters $[[6, 0, 4]]$ is generated by the generator matrix

$$\begin{pmatrix} \omega & 0 & 0 & 1 & 1 & 1 \\ 0 & \omega & 0 & \omega^2 & 1 & \omega \\ 0 & 0 & \omega & \omega^2 & \omega & 1 \\ 0 & 1 & 0 & \omega & \omega^2 & 1 \\ 0 & 0 & 1 & \omega & 1 & \omega^2 \\ 1 & \omega^2 & 0 & \omega & 0 & 0 \end{pmatrix}.$$



(a) *The "Wheel"* **(b)** *The "2-clique of 3-cliques"*

Fig. 1: *The LC Orbit of the [[6,0,4]] Hexacode*

We can transform the generator matrix into the following generator matrix of an equivalent code corresponding to a graph state,

$$\begin{pmatrix} \omega & 0 & 0 & 1 & 1 & 1 \\ 0 & \omega & 1 & 1 & 0 & 1 \\ 0 & 1 & \omega & 1 & 1 & 0 \\ 1 & 1 & 1 & \omega & 1 & 1 \\ 1 & 0 & 1 & 1 & \omega & 0 \\ 1 & 1 & 0 & 1 & 0 & \omega \end{pmatrix} = \Gamma + \omega I.$$

Γ is the adjacency matrix of the graph shown in Fig. 1a. It can also be represented by the quadratic Boolean function $f(\boldsymbol{x}) = x_0x_3 + x_0x_4 + x_0x_5 + x_1x_2 + x_1x_3 + x_1x_5 + x_2x_3 + x_2x_4 + x_3x_4 + x_3x_5$.

Recently, Glynn et al. [5, 9] has re-formulated the primitive operations that map equivalent self-dual additive codes over GF(4) to each other as a single, primitive operation on the associated graphs. This symmetry operation is referred to as *vertex neighbourhood complementation* (VNC). It was also discovered independently by Hein et al. [4] and by Van den Nest et al. [10]. The identification of this problem as a question of establishing the *local unitary equivalence* between those quantum states that can be represented as graphs or Boolean functions was presented by Parker and Rijmen at SETA 2001 [11]. Graphical representations have also been identified in the context of quantum codes by Schlingemann and Werner [7] and by Grassl et al. [8]. VNC is another name for *local complementation* (LC), referred to in the context of *isotropic systems* by Bouchet [12, 13]. LC is defined as follows.

Definition 2. Given a graph $G = (V, E)$ and a vertex $v \in V$. Let $N_v \subset V$ be the neighbourhood of v, i.e., the set of vertices adjacent to v.

The subgraph induced by N_v is complemented to obtain the LC image $G * v$. It is easy to verify that $G * v * v = G$.

Theorem 3 ([5, 9])**.** *Two graphs G and G' correspond to equivalent self-dual additive codes over* GF(4) *iff there is a finite sequence of vertices $v_1, v_2, \ldots, v_s$, such that $G * v_1 * v_2 * \cdots * v_s$ is isomorphic to G'.*

The symmetry rule can also be described in terms of quadratic Boolean functions.

Definition 4. If the quadratic monomial $x_i x_j$ occurs in the algebraic normal form of the quadratic Boolean function f, then x_i and x_j are mutual neighbours in the graph represented by f, as described by the $n \times n$ symmetric adjacency matrix Γ, where $\Gamma_{i,j} = \Gamma_{j,i} = 1$ if $x_i x_j$ occurs in f, and $\Gamma_{i,j} = 0$ otherwise. The quadratic Boolean functions f and f' are *LC equivalent* if

$$f'(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{\substack{j,k \in N_a \\ j<k}} x_j x_k \pmod 2,$$

where $a \in \mathbb{Z}_n$ and N_a comprises the neighbours of x_a in the graph representation of f.

A finite number of repeated applications of the LC operation generates the orbit classes presented in this paper and, therefore, induces an equivalence between quadratic Boolean functions. We henceforth refer to this equivalence as *LC-equivalence* and the associated orbits as *LC orbits*. If the graph representations of two self-dual additive codes over GF(4) are isomorphic, they are also considered to be equivalent. This corresponds to a permutation of the labels of the vertices in the graph or the variables in the Boolean function. We only count members of an LC orbit up to isomorphism. As an example, Fig. 1 shows the graph representation of the two only non-isomorphic members in the orbit of the $[[6, 0, 4]]$ *Hexacode*.

A recursive algorithm, incorporating the package *nauty* [14] to check for graph isomorphism, was used to generate the LC orbits enumerated in Table 1. Only the LC orbits of indecomposable codes (corresponding to connected graphs) were generated, since all decomposable codes (corresponding to unconnected graphs) can easily be constructed by combining indecomposable codes of shorter lengths.

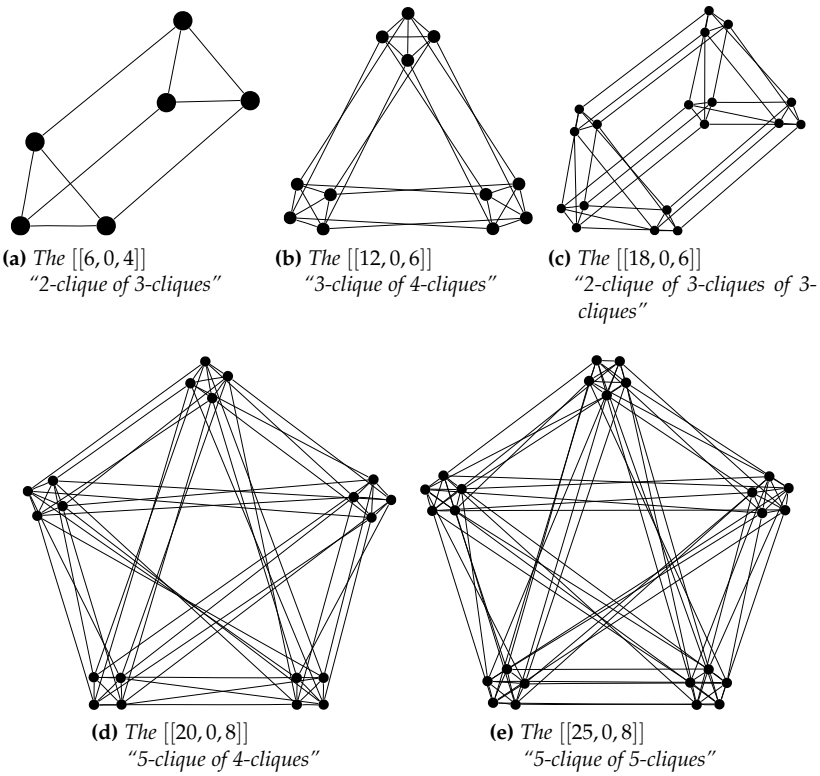
Consider, (a) self-dual additive codes over GF(4) of length n, (b) pure quantum states of n qubits which are joint eigenvectors of a

commuting set of operators from the Pauli Group [1], (c) quadratic Boolean functions of n variables, (d) undirected graphs on n vertices. Then, under a suitable interpretation, we consider objects (a), (b), (c), and (d) to be mathematically identical.

3 Regular Graph Structures

Although a number of constructions for self-dual additive codes over GF(4) exist [5, 15], it appears that the underlying symmetry of their associated graphs has not been identified or exploited to any great extent. We highlight the regularity and structure of some graphs that correspond to self-dual additive codes over GF(4) with high minimum distance. Of particular interest are the highly regular "nested clique" graphs. Fig. 2 shows a few examples of such graphs. There is an upper bound on the possible minimum distance of self-dual additive codes over GF(4) [2]. Codes that meet this bound are called *extremal*. Other bounds on the minimum distance also exist [1, 16]. Of the codes corresponding to graphs shown in Fig. 2, the $[[6,0,4]]$, $[[12,0,6]]$, and $[[20,0,8]]$ codes are extremal. To find the "nested clique" graph representations, one may search through the appropriate LC orbits. Also note that all "nested clique" graphs we have identified so far have *circulant* adjacency matrices. An exhaustive search of all graphs with circulant adjacency matrices of up to 30 vertices has been performed.

If d is the minimum distance of a self-dual additive code over GF(4), then every vertex in the corresponding graph must have a vertex degree of at least $d-1$. This follows from the fact that a vertex with degree δ corresponds to a row in the generator matrix, and therefore a codeword, of weight $\delta+1$. All the graphs shown in Fig. 2 satisfy the minimum possible regular vertex degree for the given minimum distance. Some extremal self-dual additive codes over GF(4) do not have any regular graph representation, for example the $[[11,0,5]]$ and $[[18,0,8]]$ codes. For codes of length above 25 and minimum distance higher than 8 the graph structures get more complicated. For example, with a non-exhaustive search, we did not find a graph representation of a $[[30,0,12]]$ code with regular vertex degree lower than 15.



(a) *The* $[[6, 0, 4]]$ *"2-clique of 3-cliques"*

(b) *The* $[[12, 0, 6]]$ *"3-clique of 4-cliques"*

(c) *The* $[[18, 0, 6]]$ *"2-clique of 3-cliques of 3-cliques"*

(d) *The* $[[20, 0, 8]]$ *"5-clique of 4-cliques"*

(e) *The* $[[25, 0, 8]]$ *"5-clique of 5-cliques"*

Fig. 2: *"Nested Clique" Graphs*

4 The $\{I, H, N\}^n$ Transform

LC-equivalence between two graphs can be interpreted as an equivalence between the generalised Fourier spectra of the two associated Boolean functions.

Definition 5. Let

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad N = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix},$$

where $i^2 = -1$, be the identity, Hadamard, and Negahadamard kernels, respectively.

These are *unitary* matrices, i.e., $II^\dagger = HH^\dagger = NN^\dagger = I$, where $\dagger$ means *conjugate transpose*. Let f be a Boolean function of n variables and $\boldsymbol{s} = 2^{-\frac{n}{2}}(-1)^{f(\boldsymbol{x})}$ be a vector of length 2^n. Let s_j, where $j \in \mathbb{Z}_{2^n}$, be the j-th coordinate of $\boldsymbol{s}$. Let $U = U_0 \otimes U_1 \otimes \cdots \otimes U_{n-1}$ where $U_k \in \{I, H, N\}$, and $\otimes$ is the *tensor product* (or *Kronecker product*) defined as

$$A \otimes B = \begin{pmatrix} a_{00}B & a_{01}B & \cdots \\ a_{10}B & a_{11}B & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}.$$

Let $\boldsymbol{S} = U\boldsymbol{s}$ for any of the 3^n valid choices of the $2^n \times 2^n$ transform U. Then the set of 3^n vectors, $\boldsymbol{S}$, is a multispectra with respect to the transform set, U, with $3^n 2^n$ spectral points. We refer to this multispectra as the spectrum with respect to the $\{I, H, N\}^n$ transform. (Using a similar terminology, the spectrum with respect to the $\{H\}^n$ transform would simply be the well-known Walsh-Hadamard spectrum). It can be shown that the $\{I, H, N\}^n$ spectrum of an LC orbit is invariant to within coefficient permutation. Moreover if, for a specific choice of U, $\boldsymbol{S}$ is flat (i.e., $|S_i| = |S_j|, \forall i, j$), then we can write $\boldsymbol{S} = v^{4f'(\boldsymbol{x})+h(\boldsymbol{x})}$, where f' is a Boolean function, h is any function from $\mathbb{Z}_2^n$ to $\mathbb{Z}_8$, and $v^4 = -1$. If the algebraic degree of $h(\boldsymbol{x})$ is ≤ 1, we can always eliminate $h(\boldsymbol{x})$ by post-multiplication by a tensor product of matrices from $\mathcal{D}$, the set of 2×2 diagonal and anti-diagonal unitary matrices [17], an operation that will never change the spectral coefficient magnitudes. Let M be the multiset of f' existing within the $\{I, H, N\}^n$ spectrum for the subcases where $h(\boldsymbol{x})$ is of algebraic degree ≤ 1. The $\{I, H, N\}^n$-*orbit* of f is then the set of distinct members of M. In particular, if f is quadratic then the $\{I, H, N\}^n$-orbit is the LC orbit [17].

Example 6. We look at the function $f(\boldsymbol{x}) = x_0x_1 + x_0x_2$. The corresponding bipolar vector, ignoring the normalization factor, is

$$\boldsymbol{s} = (-1)^{f(\boldsymbol{x})} = (1, 1, 1, -1, 1, -1, 1, 1)^T.$$

We choose the transform $U = N \otimes I \otimes I$ and get the result

$$\boldsymbol{S} = U\boldsymbol{s} = (v, v^7, v^7, v, v^7, v, v, v^7)^T, \quad v^4 = -1.$$

We observe that $|S_i| = 1$, $\forall i$, which means that $\boldsymbol{S}$ is flat and can be expressed as

$$\boldsymbol{S} = v^{4(x_0x_1+x_0x_2+x_1x_2)+(6x_0+6x_1+6x_2+1)}.$$

We observe that $h(\boldsymbol{x})$, the terms that are not divisible by 4, are all linear or constant. We can therefore eliminate $h(\boldsymbol{x})$, in this case by using the transform

$$D = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \otimes \begin{pmatrix} v^7 & 0 \\ 0 & v \end{pmatrix}.$$

We get the result

$$D\boldsymbol{S} = (-1)^{x_0x_1+x_0x_2+x_1x_2},$$

and thus $f'(\boldsymbol{x}) = x_0x_1 + x_0x_2 + x_1x_2$. The functions f and f' are in the same $\{I, H, N\}^n$ orbit, and since they are quadratic functions, the same LC orbit. This can be verified by applying the LC operation to the vertex corresponding to the variable x_0 in the graph representation of either function.

5 Peak-to-Average Power Ratio w.r.t. $\{I, H, N\}^n$

Definition 7. The peak-to-average power ratio of a vector, $\boldsymbol{s}$, with respect to the $\{I, H, N\}^n$ transform [18] is

$$\mathrm{PAR}_{IHN}(\boldsymbol{s}) = 2^n \max_{\substack{U \in \{I,H,N\}^n \\ k \in \mathbb{Z}_{2^n}}} |S_k|^2, \quad \text{where } \boldsymbol{S} = U\boldsymbol{s}.$$

If a vector, $\boldsymbol{s}$, has a completely flat $\{I, H, N\}^n$ spectrum (which is impossible) then $\mathrm{PAR}_{IHN}(\boldsymbol{s}) = 1$. If $\boldsymbol{s} = 2^{-\frac{n}{2}}(1, 1, \ldots, 1, 1)$ then $\mathrm{PAR}_{IHN}(\boldsymbol{s}) = 2^n$. A typical vector, $\boldsymbol{s}$, will have a $\mathrm{PAR}_{IHN}(\boldsymbol{s})$ somewhere between these extremes. For quadratic functions, PAR_{IHN} will

Table 3: *Number of LC Orbits with Length n and PAR_{IHN} p*

$p\backslash n$	1	2	3	4	5	6	7	8	9	10	11	12
2	1	1										
4			1	1	1	1						
8				1	2	5	6	9	2	1		
16					1	4	14	52	156	624	3184	12 323
32						1	5	32	212	1753	25 018	834 256
64							1	7	60	639	10 500	380 722
128								1	9	103	1578	43 013
256									1	11	163	3488
512										1	13	249
1024											1	16
2048												1

always be a power of 2. The PAR of $\boldsymbol{s}$ can be alternatively expressed in terms of the *generalised nonlinearity* [18],

$$\gamma(f) = 2^{\frac{n}{2}-1}\left(2^{\frac{n}{2}} - \sqrt{\mathrm{PAR}_{IHN}(\boldsymbol{s})}\right),$$

but in this paper we use the PAR measure. Let $\boldsymbol{s} = 2^{-\frac{n}{2}}(-1)^{f(\boldsymbol{x})}$, as before. When we talk about the PAR_{IHN} of f or its associated graph G, we mean $\mathrm{PAR}_{IHN}(\boldsymbol{s})$. It is desirable to find Boolean functions with high generalised nonlinearity and therefore low PAR_{IHN} [19]. PAR_{IHN} is an invariant of the $\{I, H, N\}^n$ orbit and, in particular, the LC orbit. We observe that Boolean functions from LC orbits associated with self-dual additive codes over GF(4) with high minimum distance typically have low PAR_{IHN}. This is not surprising as the minimum distance of a quantum code has been shown to be equal to the recently defined *aperiodic propagation criteria distance* (APC distance) [19] of the associated quadratic Boolean function, and APC is derived from the aperiodic autocorrelation which is, in turn, the autocorrelation "dual" of the spectra with respect to $\{I, H, N\}^n$. Table 3 shows PAR_{IHN} values for every LC orbit representative for $n \leq 12$.

Definition 8. Let $\alpha(G)$ be the independence number of a graph G, i.e., the size of the maximum independent set in G. Let $[G]$ be the set of all graphs in the LC orbit of G. We then define $\lambda(G) = \max_{H\in[G]} \alpha(H)$, i.e., the size of the maximum independent set over all graphs in the LC orbit of G.

Table 4: *Range of λ for Codes of Length n and Distance d*

$d\backslash n$	2	3	4	5	6	7	8	9	10	11	12
2	**1**	**2**	**2**,3	3,4	3–5	3–6	3–7	4–8	4–9	4–10	4–11
3				**2**	3	**3**,4	3,4	3–5	4–6	4–7	4–8
4					**2**		**3**,4	**3**,4	**3**–5	4–6	4–7
5										**4**	4
6											**4**

Consider as an example the Hexacode which has two non-isomorphic graphs in its orbit (Fig. 1). It is evident that the size of the largest independent set of each graph is 2, so $\lambda = 2$. The values of λ for all LC orbits for $n \leq 12$ clearly show that λ and d, the minimum distance of the associated self-dual additive code over GF(4), are related. LC orbits associated with codes with high minimum distance typically have small values for λ. Table 4 summarises this observation by giving the ranges of λ observed for all LC orbits associated with codes of given lengths and minimum distances. For instance, $[[12, 0, 2]]$ codes exist with any value of λ between 4 and 11, while $[[12, 0, 5]]$ and $[[12, 0, 6]]$ codes only exist with $\lambda = 4$.

Definition 9. Let Λ_n be the minimum value of λ over all LC orbits with n vertices.

From Table 4 we observe that $\Lambda_n = 2$ for n from 3 to 6, $\Lambda_n = 3$ for n from 7 to 10, and $\Lambda_n = 4$ when n is 11 or 12.

Theorem 10. $\Lambda_{n+1} \geq \Lambda_n$, *i.e.,* Λ_n *is monotonically nondecreasing when the number of vertices is increasing.*

Proof. Consider a graph $G = (V, E)$ with $n + 1$ vertices. Select a vertex v and let G' be the induced subgraph on the n vertices $V\backslash\{v\}$. We generate the LC-orbit of G'. The LC operations may add or remove edges between G' and v, but the presence of v does not affect the LC orbit of G'. The size of the largest independent set in the LC orbit of G' is at least Λ_n. This is also an independent set in the LC orbit of G, so $\Lambda_{n+1} \geq \Lambda_n$. □

A very loose lower bound on Λ_n can also be given. Consider a graph containing a clique of size k. It is easy to see that an LC operation on any vertex in the clique will produce an independent set of size $k - 1$.

Table 5: *Upper Bounds on* Λ_n

n	13	14	15	16	17	18	19	20	21
$\Lambda_n \leq$	4	4	5	5	5	6	6	6	9

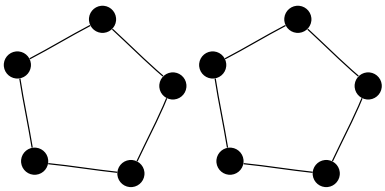


Fig. 3: *The "Double 5-Cycle" Graph*

Thus the maximum clique in an LC orbit, where the largest independent set has size λ, can not be larger than $\lambda + 1$. If r is the *Ramsey number* $R(k, k+1)$ [20], then it is guaranteed that all simple undirected graphs with minimum r vertices will have either an independent set of size k or a clique of size $k+1$. It follows that all LC orbits with at least r vertices must have $\lambda \geq k$. Thus $\Lambda_n \geq k$ for $n \geq r$. For instance, $R(3,4) = 9$, so LC orbits with at least 9 vertices can not have λ smaller than 3.

For $n > 12$, we have computed the value of λ for some graphs corresponding to self-dual additive codes over GF(4) with high minimum distance. This gives us upper bounds on the value of Λ_n, as shown in Table 5. The bounds on Λ_{13} and Λ_{14} are tight, since $\Lambda_{12} = 4$ and $\Lambda_{n+1} \geq \Lambda_n$.

For $n = 10$, there is a unique LC orbit that satisfies, optimally, $\lambda = 3$, $\mathrm{PAR}_{IHN} = 8$ and $d = 4$. One of the graphs in this orbit is the *graph complement* of the "double 5-cycle" graph, shown in Fig. 3.

Theorem 11 ([11]). *Given a graph $G = (V, E)$ with a maximum independent set $A \subset V$, $|A| = \alpha(G)$. Let $\boldsymbol{s} = (-1)^{f(\boldsymbol{x})}$, where $f(\boldsymbol{x})$ is the Boolean function representation of G. Let $U = \bigotimes_{i \in A} H_i \bigotimes_{i \notin A} I_i$, i.e., the transform applying H to variables corresponding to vertices $v \in A$ and I to all other variables. Then* $\max_{k \in \mathbb{Z}_{2^n}} |S_k|^2 = 2^{\alpha(G)}$, *where $\boldsymbol{S} = U\boldsymbol{s}$.*

Arratia et al. [21] introduced the *interlace polynomial* $q(G)$ of a graph G. Aigner and van der Holst [22] later introduced the interlace polynomial $Q(G)$. Riera and Parker [23] showed that $q(G)$ is related to the $\{I, H\}^n$

spectra of the quadratic Boolean function corresponding to G, and that $Q(G)$ is related to the $\{I, H, N\}^n$ spectra.

Theorem 12 ([23])**.** *Let f be a quadratic Boolean function and G its associated graph. Then PAR_{IHN} of f is equal to $2^{\deg(Q(G))}$, where $\deg(Q(G))$ is the degree of the interlace polynomial $Q(G)$.*

Theorem 13. *If the maximum independent set over all graphs in the LC orbit $[G]$ has size $\lambda(G)$, then all functions corresponding to graphs in the orbit will have $PAR_{IHN} = 2^{\lambda(G)}$.*

Proof. For brevity, let $P(G) = \mathrm{PAR}_{IHN}(\boldsymbol{s})$, where $\boldsymbol{s} = 2^{-\frac{n}{2}}(-1)^{f(\boldsymbol{x})}$, and $f(\boldsymbol{x})$ is the Boolean function representation of G. From Theorem 11 it follows that $P(G) \geq 2^{\lambda(G)}$. Choose $H = (V, E) \in [G]$ with $\alpha(H) = \lambda(G)$. If $|V| = 1$ or 2, the theorem is true. We will prove the theorem for $n > 2$ by induction on $|V|$. We will show that $P(H) \leq 2^{\alpha(H)}$, which is equivalent to saying that $P(G) \leq 2^{\lambda(G)}$. It follows from Theorem 12 and the definition of $Q(H)$ by Aigner and van der Holst [22] that $P(H) = \max\{P(H \setminus u), P(H * u \setminus u), P(H * u * v * u \setminus u)\}$. (We recall that $H * u$ denotes the LC operation on vertex u of H.) Assume, by induction hypothesis, that $P(H \setminus u) = 2^{\lambda(H \setminus u)}$. Therefore, $P(H \setminus u) = 2^{\alpha(K \setminus u)}$ for some $K \setminus u \in [H \setminus u]$. Note that $K \setminus u \in [H \setminus u]$ implies $K \in [H]$. It must then be true that $\alpha(K \setminus u) \leq \alpha(K) \leq \alpha(H)$, and it follows that $P(H \setminus u) \leq 2^{\alpha(H)}$. Similar arguments hold for $P(H * u \setminus u)$ and $P(H * u * v * u \setminus u)$, so $P(H) \leq 2^{\alpha(H)}$. □

As an example, the Hexacode has $\lambda = 2$ and therefore $\mathrm{PAR}_{IHN} = 2^2 = 4$.

Corollary 14. *Any quadratic Boolean function of n or more variables must have $PAR_{IHN} \geq 2^{\Lambda_n}$.*

Definition 15. PAR_{IH} is the peak-to-average power ratio with respect to the transform set $\{I, H\}^n$, otherwise defined in the same way as PAR_{IHN}.

Definition 16. PAR_U is the peak-to-average power ratio with respect to the infinite transform set $\{U\}^n$, consisting of matrices of the form

$$U = \begin{pmatrix} \cos\theta & \sin\theta e^{i\phi} \\ \sin\theta & -\cos\theta e^{i\phi} \end{pmatrix},$$

where $i^2 = -1$, and θ and ϕ can take any real values. $\{U\}$ comprises all 2×2 unitary transforms to within a post-multiplication by a matrix from $\mathcal{D}$, the set of 2×2 diagonal and anti-diagonal unitary matrices.

Theorem 17 ([11]). *If $\boldsymbol{s}$ corresponds to a bipartite graph, then $PAR_U(\boldsymbol{s}) = PAR_{IH}(\boldsymbol{s})$.*

It is obvious that $\{I,H\}^n \subset \{I,H,N\}^n \subset \{U\}^n$, and therefore that $\mathrm{PAR}_{IH} \leq \mathrm{PAR}_{IHN} \leq \mathrm{PAR}_U$. We then get the following corollary of Theorems 13 and 17.

Corollary 18. *If an LC orbit, $[G]$, contains a bipartite graph, then all functions corresponding to graphs in the orbit will have $PAR_U = 2^{\lambda(G)}$.*

Thus, all LC orbits with a bipartite member have $\mathrm{PAR}_{IHN} = \mathrm{PAR}_U$. Note that these orbits will always have $\mathrm{PAR}_U \geq 2^{\lceil \frac{n}{2} \rceil}$ [11] and that the fraction of LC orbits which have a bipartite member appears to decrease exponentially as the number of vertices increases. In the general case, PAR_{IHN} is only a lower bound on PAR_U. For example, the Hexacode has $\mathrm{PAR}_{IHN} = 4$, but a tighter lower bound on PAR_U is 4.486 [11]. (This bound has later been improved to 5.103 [24].)

6 Construction for Low PAR_{IHN}

So far we have only considered *quadratic* Boolean functions which correspond to graphs and self-dual additive codes over GF(4). For cryptographic purposes, we are interested in Boolean functions of degree higher than two. Such functions can be represented by *hypergraphs*, but they do not correspond to quantum stabilizer codes or self-dual additive codes over GF(4). A nonquadratic Boolean function, $f(\boldsymbol{x})$, can, however, be interpreted as a quantum state described by the probability distribution vector $\boldsymbol{s} = 2^{-\frac{n}{2}}(-1)^{f(\boldsymbol{x})}$. A single quantum state corresponds to a quantum code of dimension zero whose minimum distance is the APC distance [19]. The APC distance is the weight of the minimum weight quantum error operator that gives an errored state not orthogonal to the original state and therefore not guaranteed to be detectable.

We are interested in finding Boolean functions of algebraic degree greater than two with low PAR_{IHN}, but exhaustive searching becomes infeasible with more than a few variables. We therefore propose a construction technique for nonquadratic Boolean functions with low PAR_{IHN} using the best quadratic functions as building blocks. Before we describe our construction we must first state what we mean by "low PAR_{IHN}". For $n = 6$ to $n = 10$ we computed PAR_{IHN} for samples from the space $\mathbb{Z}_2^{2^n}$, to determine the range of PAR_{IHN} we can expect just

Table 6: *Sampled Range of PAR_{IHN} for n from 6 to 10*

n	Samples	Range of PAR_{IHN}
6	50 000	6.5–25.0
7	20 000	9.0–28.125
8	5 000	12.25–28.125
9	2 000	14.0625–30.25
10	1 000	18.0–34.03

by guessing. Table 6 summarises these results. If we can construct Boolean functions with PAR_{IHN} lower than the sampled minimum, we can consider our construction to be somewhat successful.

Parker and Tellambura [25, 26] proposed a generalisation of the Maiorana-McFarland construction for Boolean functions that satisfies a tight upper bound on PAR with respect to the $\{H, N\}^n$ transform (and other transform sets), this being a form of Golay complementary set construction and a generalisation of the construction of Rudin and Shapiro and of Davis and Jedwab [27]. Let $p(\boldsymbol{x})$ be a Boolean function of $n = \sum_{j=0}^{L-1} t_j$ variables, where $T = \{t_0, t_1, \ldots, t_{L-1}\}$ is a set of positive integers and $\boldsymbol{x} \in \mathbb{Z}_2^n$. Let $\boldsymbol{y_j} \in \mathbb{Z}_2^{t_j}$, $0 \leq j < L$, such that $\boldsymbol{x} = \boldsymbol{y_0} \times \boldsymbol{y_1} \times \cdots \times \boldsymbol{y_{L-1}}$. Construct $p(\boldsymbol{x})$ as follows.

$$p(\boldsymbol{x}) = \sum_{j=0}^{L-2} \theta_j(\boldsymbol{y_j})\gamma_j(\boldsymbol{y_{j+1}}) + \sum_{j=0}^{L-1} g_j(\boldsymbol{y_j}), \tag{1}$$

where θ_j is a permutation: $\mathbb{Z}_2^{t_j} \to \mathbb{Z}_2^{t_{j+1}}$, γ_j is a permutation: $\mathbb{Z}_2^{t_{j+1}} \to \mathbb{Z}_2^{t_j}$, and g_j is any Boolean function of t_j variables. It has been shown [26] that the function $p(\boldsymbol{x})$ will have $PAR_{HN} \leq 2^{t_{\max}}$, where $t_{\max}$ is the largest integer in T. It is helpful to visualise this construction graphically, as in Fig. 4. In this example, the size of the largest partition is 3, so $PAR_{HN} \leq 8$, regardless of what choices we make for θ_j, γ_j, and g_j.

Observe that if we set $L = 2$, $t = t_0 = t_1$, let θ_0 be the identity permutation, and $g_0 = 0$, Eq. (1) reduces to the Maiorana-McFarland construction over $2t$ variables. Eq. (1) can also be viewed as a generalisation of the "path graph", $f(\boldsymbol{x}) = x_0x_1 + x_1x_2 + \cdots + x_{n-2}x_{n-1}$, which has optimal PAR with respect to $\{H, N\}^n$. Unfortunately, the "path graph" is not a particularly good construction for low PAR_{IHN}. But as we have seen, graphs corresponding to self-dual additive codes

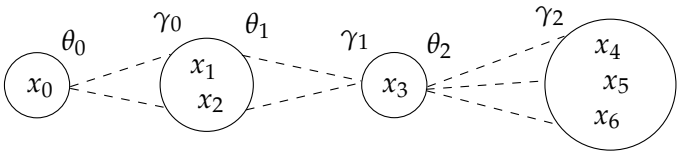


Fig. 4: *Example of Construction with* $PAR_{HN} \leq 8$

over GF(4) with high minimum distance do give us Boolean functions with low PAR_{IHN}. We therefore propose the following generalised construction.

$$p(\boldsymbol{x}) = \sum_{i=0}^{L-1} \sum_{j=i+1}^{L-1} \Gamma_{i,j}(\boldsymbol{y_i}) \Gamma_{j,i}(\boldsymbol{y_j}) + \sum_{j=0}^{L-1} g_j(\boldsymbol{y_j}), \tag{2}$$

where $\Gamma_{i,j}$ is either a permutation: $\mathbb{Z}_2^{t_i} \rightarrow \mathbb{Z}_2^{t_j}$, or $\Gamma_{i,j} = 0$, and g_j is any Boolean function of t_j variables. It is evident that Γ can be thought of as a "generalised adjacency matrix", where the entries, $\Gamma_{i,j}$, are no longer 0 or 1 but, instead, 0 or permutations from $\mathbb{Z}_2^{t_i}$ to $\mathbb{Z}_2^{t_j}$. Eq. (1) then becomes a special case where $\Gamma_{i,j} = 0$ except for when $j = i + 1$ (i.e., the "generalised adjacency matrix" of the "path graph"). In order to minimise PAR_{IHN} we choose the form of the matrix Γ according to the adjacency matrix of a self-dual additive code over GF(4) with high minimum distance. We also choose the "offset" functions, g_j, to be Boolean functions corresponding to self-dual additive codes over GF(4) with high minimum distance. Finally for the non-zero $\Gamma_{i,j}$ entries, we choose selected permutations, preferably nonlinear to increase the overall degree. Here are some initial results which demonstrate that, using Eq. (2), we can construct Boolean functions of algebraic degree greater than 2 with low PAR_{IHN}. (We use an abbreviated ANF notation for some many-term Boolean functions, e.g. $012, 12, 0$ is short for $x_0x_1x_2 + x_1x_2 + x_0$.)

Example 19 ($n = 8$)**.** Use the Hexacode graph $f =$ 01, 02, 03, 04, 05, 12, 23, 34, 45, 51 as a template. Let $t_0 = 3$, $t_1 = t_2 = t_3 = t_4 = t_5 = 1$.

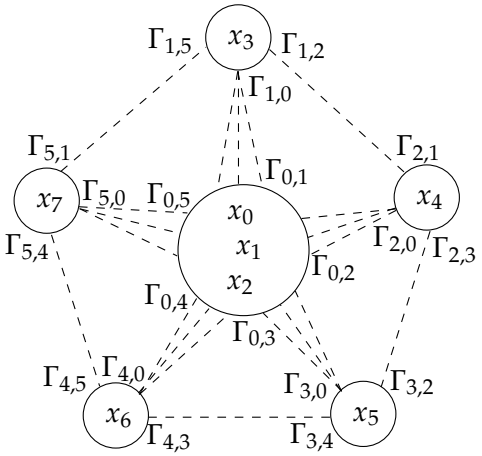


Fig. 5: *Example of Construction with Low* PAR_{IHN}

(Fig. 5.) We use the following matrix Γ.

$$\Gamma = \begin{pmatrix} 0 & 02,1 & 02,1 & 02,1 & 02,1 & 02,1 \\ 3 & 0 & 3 & 0 & 0 & 3 \\ 4 & 4 & 0 & 4 & 0 & 0 \\ 5 & 0 & 5 & 0 & 5 & 0 \\ 6 & 0 & 0 & 6 & 0 & 6 \\ 7 & 7 & 0 & 0 & 7 & 0 \end{pmatrix}$$

Let $g_0(\boldsymbol{y_0}) = 01, 02, 12$ and all other g_j any arbitrary affine functions. Then, using Eq. (2) to construct $p(\boldsymbol{x})$ we get $p(\boldsymbol{x}) = 023, 024, 025, 026, 027, 01, 02, 12, 13, 14, 15, 16, 17, 34, 37, 45, 56, 67$. Then $p(\boldsymbol{x})$ has $\mathrm{PAR}_{IHN} = 9.0$.

Example 20 ($n = 8$)**.** Use the Hexacode graph $f = 01, 02, 03, 04, 05, 12, 23, 34, 45, 51$ as a template. Let $t_0 = 3$, $t_1 = t_2 = t_3 = t_4 = t_5 = 1$. (Fig. 5.) We use the following matrix Γ.

$$\Gamma = \begin{pmatrix} 0 & 02,1 & 12,0,1,2 & 01,02,12,1,2 & 01,02,12 & 02,12,1,2 \\ 3 & 0 & 3 & 0 & 0 & 3 \\ 4 & 4 & 0 & 4 & 0 & 0 \\ 5 & 0 & 5 & 0 & 5 & 0 \\ 6 & 0 & 0 & 6 & 0 & 6 \\ 7 & 7 & 0 & 0 & 7 & 0 \end{pmatrix}$$

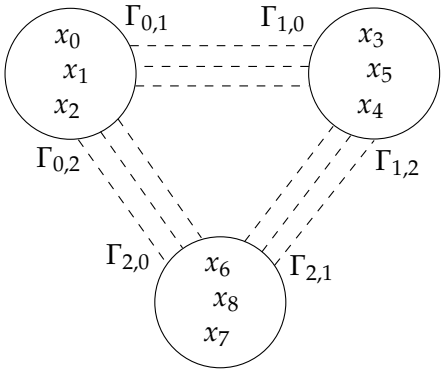


Fig. 6: *Example of Construction with Low PAR_{IHN}*

Let $g_0(\boldsymbol{y_0}) = 01, 12$ and all other g_j any arbitrary affine functions. Then, using Eq. (2) to construct $p(\boldsymbol{x})$ we get $p(\boldsymbol{x}) =$ 015, 016, 023, 025, 026, 027, 124, 125, 126, 127, 01, 04, 12, 13, 14, 15, 17, 24, 25, 27, 34, 37, 45, 56, 67. Then $p(\boldsymbol{x})$ has $\text{PAR}_{IHN} = 9.0$.

Example 21 ($n = 9$)**.** Use the triangle graph $f = 01, 02, 12$ as a template. Let $t_0 = t_1 = t_2 = 3$. (Fig. 6.) Assign the permutations

$$\begin{aligned} \Gamma_{0,1} = \Gamma_{0,2} &= (12,0,1,2)(01,2)(02,1,2), \\ \Gamma_{1,0} &= (34,5)(35,4,5)(45,3,4,5), \\ \Gamma_{1,2} &= (45,3,4,5)(34,5)(35,4,5), \\ \Gamma_{2,0} &= (68,7,8)(78,6,7,8)(67,8), \\ \Gamma_{2,1} &= (78,6,7,8)(67,8)(68,7,8). \end{aligned}$$

Let $g_0(\boldsymbol{y_0}) = 01, 02, 12$, $g_1(\boldsymbol{y_1}) = 34, 35, 45$, and $g_2(\boldsymbol{y_2}) = 67, 68, 78$. Then, using Eq. (2) to construct $p(\boldsymbol{x})$ we get, $p(\boldsymbol{x}) =$ 0135, 0178, 0245, 0267, 1234, 1268, 3467, 3568, 4578, 014, 015, 016, 017, 018, 023, 024, 025, 028, 034, 068, 125, 127, 128, 134, 145, 167, 168, 234, 235, 245, 267, 268, 278, 348, 357, 358, 378, 456, 457, 458, 468, 478, 567, 568, 578, 05, 07, 08, 13, 14, 17, 23, 25, 26, 28, 36, 37, 38, 46, 56, 58, 01, 02, 12, 34, 35, 45, 67, 68, 78. Then $p(\boldsymbol{x})$ has $\text{PAR}_{IHN} = 10.25$.

The examples of our construction satisfy a low PAR_{IHN}. Further work should ascertain the proper choice of permutations. Finally, there is an even more obvious variation of the construction given by Eq. (2), suggested by the graphs of Fig. 2, where the functions g_j are chosen

either to be quadratic cliques or to be further "nested" versions of Eq. (2). We will report on this variation in a future paper.

REFERENCES

[1] CALDERBANK, A. R., RAINS, E. M., SHOR, P. M., SLOANE, N. J. A.: Quantum error correction via codes over GF(4). *IEEE Trans. Inform. Theory* 44(4), 1369–1387, 1998. arXiv:quant-ph/9608006.

[2] RAINS, E. M., SLOANE, N. J. A.: Self-dual codes. In *Handbook of Coding Theory*, pp. 177–294. North-Holland, Amsterdam, 1998. arXiv:math.CO/0208001.

[3] HÖHN, G.: Self-dual codes over the Kleinian four group. *Math. Ann.* 327(2), 227–255, 2003. arXiv:math.CO/0005266.

[4] HEIN, M., EISERT, J., BRIEGEL, H. J.: Multi-party entanglement in graph states. *Phys. Rev. A* 69(6), 2004. arXiv:quant-ph/0307130.

[5] GLYNN, D. G., GULLIVER, T. A., MAKS, J. G., GUPTA, M. K.: The geometry of additive quantum codes, 2004. Submitted to Springer.

[6] SLOANE, N. J. A.: The On-Line Encyclopedia of Integer Sequences. `http://www.research.att.com/~njas/sequences/`.

[7] SCHLINGEMANN, D., WERNER, R. F.: Quantum error-correcting codes associated with graphs. *Phys. Rev. A* 65(1), 2002. arXiv:quant-ph/0012111.

[8] GRASSL, M., KLAPPENECKER, A., RÖTTELER, M.: Graphs, quadratic forms, and quantum codes. In *Proc. IEEE Int. Symp. Inform. Theory*, p. 45. 2002. arXiv:quant-ph/0703112.

[9] GLYNN, D. G.: On self-dual quantum codes and graphs, 2002. Submitted to Electron. J. Combin.

[10] VAN DEN NEST, M., DEHAENE, J., DE MOOR, B.: Graphical description of the action of local Clifford transformations on graph states. *Phys. Rev. A* 69(2), 2004. arXiv:quant-ph/0308151.

[11] PARKER, M. G., RIJMEN, V.: The quantum entanglement of binary and bipolar sequences. In *Sequences and Their Applications – SETA 2001*, Discrete Math. Theor. Comput. Sci., pp. 296–309. Springer, London, 2002. arXiv:quant-ph/0107106.

[12] BOUCHET, A.: Isotropic systems. *European J. Combin.* 8(3), 231–244, 1987.

[13] BOUCHET, A.: Recognizing locally equivalent graphs. *Discrete Math.* 114(1–3), 75–86, 1993.

[14] McKay, B. D.: *nauty User's Guide, Version 2.2*, 2003. `http://cs.anu.edu.au/~bdm/nauty/`.

[15] Gulliver, T. A., Kim, J.-L.: Circulant based extremal additive self-dual codes over GF(4). *IEEE Trans. Inform. Theory* 50(2), 359–366, 2004.

[16] Grassl, M.: Bounds on d_{min} for additive $[[n,k,d]]$ QECC, Feb. 2003. `http://iaks-www.ira.uka.de/home/grassl/QECC/TableIII.html`.

[17] Riera, C., Parker, M. G.: Generalised bent criteria for Boolean functions (I). *IEEE Trans Inform. Theory* 52(9), 4142–4159, 2006. arXiv:cs.IT/0502049.

[18] Parker, M. G.: Generalised S-box nonlinearity, 2003. NESSIE Public Document, NES/DOC/UIB/WP5/020/A.

[19] Danielsen, L. E., Gulliver, T. A., Parker, M. G.: Aperiodic propagation criteria for Boolean functions. *Inform. and Comput.* 204(5), 741–770, 2006.

[20] Radziszowski, S. P.: Small Ramsey numbers. *Elect. J. Combinatorics* 1, 1994. Dynamic Survey 1.

[21] Arratia, R., Bollobás, B., Sorkin, G. B.: The interlace polynomial of a graph. *J. Combin. Theory Ser. B* 92(2), 199–233, 2004. arXiv:math.CO/0209045.

[22] Aigner, M., van der Holst, H.: Interlace polynomials. *Linear Algebra Appl.* 377, 11–30, 2004.

[23] Riera, C., Parker, M. G.: One and two-variable interlace polynomials: A spectral interpretation. In *Coding and Cryptography*, *Lecture Notes in Comput. Sci.*, vol. 3969, pp. 397–411. Springer, Berlin, 2006. arXiv:cs.IT/0504102.

[24] Parker, M. G., Gulliver, T. A.: On graph symmetries and equivalence of the six variable double-clique and wheel, 2003. Unpublished.

[25] Parker, M. G., Tellambura, C.: A construction for binary sequence sets with low peak-to-average power ratio. In *Proc. IEEE Int. Symp. Inform. Theory*, p. 239. 2002.

[26] Parker, M. G., Tellambura, C.: A construction for binary sequence sets with low peak-to-average power ratio. Tech. Rep. 242, Dept. Informat., Univ. Bergen, Norway, 2003.

[27] Davis, J. A., Jedwab, J.: Peak-to-mean power control in OFDM, Golay complementary sequences and Reed-Muller codes. *IEEE Trans. Inform. Theory* 45(7), 2397–2417, 1999.

PAPER IV

APERIODIC PROPAGATION CRITERIA FOR BOOLEAN FUNCTIONS

Lars Eirik Danielsen T. Aaron Gulliver
Matthew G. Parker

APERIODIC PROPAGATION CRITERIA FOR BOOLEAN FUNCTIONS

Lars Eirik Danielsen* T. Aaron Gulliver†
Matthew G. Parker*

We characterise the aperiodic autocorrelation of a Boolean function f and define the aperiodic propagation criteria (APC) of degree l and order q. We establish a strong similarity between APC and the extended propagation criteria (EPC) as defined by Preneel et al. in 1991, although the criteria are not identical. We also show how aperiodic autocorrelation can be related to the first derivative of f. We further propose the metric APC distance and show that quantum error correcting codes are natural candidates for Boolean functions with favourable APC distance.

1. INTRODUCTION

Imagine the block cipher scenario where an attacker has knowledge of the values of a fixed subset, $\boldsymbol{\mu}$, of the plaintext bits and any subset of the ciphertext bits, for multiple plaintext/ciphertext pairs. Moreover he is able to modify any of the plaintext bits from the set $\boldsymbol{\mu}$, in order to realise a differential attack on the cipher. For a given cipher, what is the smallest size of $\boldsymbol{\mu}$ such that a biased differential can be established across the cipher? This scenario motivates us to define *aperiodic propagation*

*Department of Informatics, University of Bergen, PB 7803, N-5020 Bergen, Norway.
†Dept. of Electrical & Computer Engineering, University of Victoria, P.O. Box 3055, STN CSC, Victoria, BC, V8W 3P6 Canada.

criteria (APC) for a Boolean function such that *APC distance* is this minimum size for $\boldsymbol{\mu}$ for a constituent Boolean function of the cipher. We also define multivariate *aperiodic autocorrelation* of a Boolean function, from which APC is derived.

Now imagine a similar scenario where the attacker has knowledge of the values of a fixed subset, $\boldsymbol{\mu}$, of the plaintext bits, and he is able to modify any subset, $\boldsymbol{a}$, of the plaintext bits, but this time $\boldsymbol{a}$ is not necessarily a subset of $\boldsymbol{\mu}$. For a given cipher, and for a given size for $\boldsymbol{a}$, what is the smallest size for $\boldsymbol{\mu}$ such that a biased differential can be established across the cipher? Preneel et al. [1] have defined *extended propagation criteria* (EPC) such that, for a constituent Boolean function of the cipher, EPC(l) of order q means that a biased differential cannot be found if $\boldsymbol{\mu}$ is of size q or less given that $\boldsymbol{a}$ is of size l or less. To ease comparison with APC, we further propose *EPC distance* to be the minimum size of $\boldsymbol{\mu} \cup \boldsymbol{a}$ such that a biased differential can be found. EPC is also considered in [2, 3].

One purpose of this paper is to characterise aperiodic autocorrelation for a Boolean function, to motivate its use for cryptanalysis, and to consider constructions for Boolean functions with favourable aperiodic criteria, where favourable here means that the aperiodic coefficients are zero at low weight indices. Preneel et al. [1] propose (periodic) *propagation criteria* (PC) of degree l and order q which evaluates periodic properties of a Boolean function when q of the input bits are kept constant. In the same way we propose *aperiodic propagation criteria* (APC) of degree l and order q to evaluate aperiodic properties when q bits are kept constant. It is then natural to compare APC with EPC.

By interpreting our Boolean function of m variables as a quantum state of m qubits, we also establish, rather surprisingly, that the APC distance of a quadratic Boolean function is equal to the minimum distance of an associated zero-dimensional *quantum error-correcting code* (QECC) which represents, in turn, a highly entangled pure quantum state [4]. We apply recent results on quantum codes to the construction of quadratic Boolean functions with favourable APC. This suggests that the disciplines of *quantum entanglement* and cryptographic criteria for Boolean functions are closely related [5]. The mapping of Boolean functions into Hilbert space allows one to apply *local unitary transforms* to establish orbits of Boolean functions over which APC distance is invariant. Orbits of quadratic functions can be generated by successive *local complementation* (LC) operations on associated graphs [6–9]. These graph operations encode the action of a special subset of the

local unitary transforms. Similarly, APC distance-invariant orbits of functions of algebraic degree greater than two can also be generated by application of the same set of local unitary transforms. Therefore, a second purpose of this paper is to re-cast the construction of QECCs as a problem of construction of Boolean functions. As a result, we are able to generalise the set of QECCs to Boolean functions of degree greater than two, whereas conventional QECCs only map to Boolean functions of degree two.

This paper is structured as follows. After establishing the notation, we characterise the aperiodic and fixed-aperiodic autocorrelation for a Boolean function. We then define APC, elaborate on the similarities between APC and EPC, and define APC and EPC distance metrics. We consider constructions for quadratic Boolean functions with favourable APC, using known results for QECCs. We also highlight the unusual LC symmetry. Finally we consider the challenging problem of finding constructions for Boolean functions of algebraic degree greater than two with favourable APC, and we describe the generalisation of LC for such functions. We also show, in Appendix B, how to use aperiodic coefficients to compute the combined periodic/negaperiodic coefficients, and vice versa. Symmetries associated with aperiodic autocorrelation are described in Appendix C. Finally Appendix D presents the results of the (truncated) differential analysis of a few state-of-the-art S-boxes with respect to periodic, aperiodic, and fixed-aperiodic autocorrelation.

2. PRELIMINARIES

Let $\mathcal{B}_m$ denote the set of all Boolean functions on m variables. For $\boldsymbol{a} = (a_0, a_1, \ldots, a_{m-1}) \in \mathrm{GF}(2)^m$, the *Hamming weight* of $\boldsymbol{a}$ is

$$\mathrm{wt}(\boldsymbol{a}) = \sum_{i=0}^{m-1} a_i. \tag{1}$$

We define the operators $^- : \mathrm{GF}(2)^m \to \mathrm{GF}(2)^m$ and $\& : \mathrm{GF}(2)^m \times \mathrm{GF}(2)^m \to \mathrm{GF}(2)^m$ as bitwise negation and modular multiplication modulo 2, respectively. Let $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathrm{GF}(2)^m$, then

$$\boldsymbol{c} = \boldsymbol{a} \,\&\, \boldsymbol{b} \quad \Rightarrow \quad c_i = a_i b_i,\ 0 \le i < m. \tag{2}$$

$$\boldsymbol{c} = \overline{\boldsymbol{a}} \quad \Rightarrow \quad c_i = a_i + 1,\ 0 \le i < m. \tag{3}$$

Let $\boldsymbol{a}, \boldsymbol{b} \in \mathrm{GF}(2)^m$, then

$$\boldsymbol{b} \preceq \boldsymbol{a} \quad \Leftrightarrow \quad b_i \le a_i,\ \ 0 \le i < m, \tag{4}$$

and we say that $\boldsymbol{a}$ *covers* $\boldsymbol{b}$.

The *dual*, $V^{\perp}$, of a subspace $V \subset \mathrm{GF}(2)^m$ can be described relative to the scalar product,

$$V^{\perp} = \{\boldsymbol{x} \in \mathrm{GF}(2)^m \mid \boldsymbol{x} \cdot \boldsymbol{y} = 0,\ \boldsymbol{y} \in V\}. \tag{5}$$

In particular, for $\boldsymbol{r} \in \mathrm{GF}(2)^m$, we define $V_{\boldsymbol{r}}$ as

$$V_{\boldsymbol{r}} = \{\boldsymbol{x} \in \mathrm{GF}(2)^m \mid \boldsymbol{x} \preceq \boldsymbol{r}\}. \tag{6}$$

Moreover, for any $\boldsymbol{k} \in \mathrm{GF}(2)^m$, $\boldsymbol{k} + V$ defines a *coset* of V.

Let E be any subset of $\mathrm{GF}(2)^m$. For any $f \in \mathcal{B}_m$ we define $f\phi_E$ as the *restriction* of f to E such that $f\phi_E(\boldsymbol{x}) = 1$ iff $f(\boldsymbol{x}) = 1$ and $\boldsymbol{x} \in E$. If E is a k-dimensional linear subspace of $\mathrm{GF}(2)^m$ then, for any coset, $\boldsymbol{b} + E$, we identify $f\phi_{\boldsymbol{b}+E}$ with a Boolean function in $\mathcal{B}_k$, where the function obtained depends on $\boldsymbol{b}$.

For any $f \in \mathcal{B}_m$ we define $\mathcal{F}(f)$ as

$$\mathcal{F}(f) = \sum_{\boldsymbol{x} \in \mathrm{GF}(2)^m} (-1)^{f(\boldsymbol{x})}. \tag{7}$$

If E is a k-dimensional linear subspace of $\mathrm{GF}(2)^m$ then, for any coset $\boldsymbol{b} + E$,

$$\mathcal{F}(f\phi_{\boldsymbol{b}+E}) = \sum_{\boldsymbol{x} \in \boldsymbol{b}+E} (-1)^{f(\boldsymbol{x})}. \tag{8}$$

The (Walsh-Hadamard) *Fourier spectrum* of $f \in \mathcal{B}_m$ is expressed as the multi-set

$$\{\mathcal{F}(f + \boldsymbol{\alpha} \cdot \boldsymbol{x}),\ \boldsymbol{\alpha} \in \mathrm{GF}(2)^m\}. \tag{9}$$

Definition 1. Let $f \in \mathcal{B}_m$ and let t be some positive integer. The function f is said to be *correlation-immune* of order t if and only if $\mathcal{F}(f + \boldsymbol{\alpha} \cdot \boldsymbol{x}) = 0$ for any $\boldsymbol{\alpha} \in \mathrm{GF}(2)^m$ such that $1 \leq \mathrm{wt}(\boldsymbol{\alpha}) \leq t$. Moreover, if such an f is also balanced, it is said to be *t-resilient*. A *balanced* function with no correlation-immunity is 0-resilient.

For any $f \in \mathcal{B}_m$ and $\boldsymbol{a} \in \mathrm{GF}(2)^m$, the *first derivative* of f with respect to $\boldsymbol{a}$ is given by $\mathcal{D}_{\boldsymbol{a}} f \in \mathcal{B}_m$, where

$$\mathcal{D}_{\boldsymbol{a}} f = f(\boldsymbol{x}) + f(\boldsymbol{x} + \boldsymbol{a}). \tag{10}$$

In the sequel we use expressions of the form $\mathcal{D}_{\boldsymbol{a}} f \phi_E$ which should always be taken to mean $(\mathcal{D}_{\boldsymbol{a}} f)\phi_E$, i.e., we omit brackets for clarity.

For $\boldsymbol{a}, \boldsymbol{k}, \boldsymbol{\mu} \in \mathrm{GF}(2)^m$, $\boldsymbol{a} \preceq \overline{\boldsymbol{\mu}}$, $\boldsymbol{k} \preceq \boldsymbol{\mu}$, the *fixed-periodic autocorrelation* coefficients, $p_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}$, of f after fixing the subspace $V_{\boldsymbol{\mu}}$ to $\boldsymbol{k}$, can be defined by

$$p_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}} = \mathcal{F}(\mathcal{D}_{\boldsymbol{a}} f \phi_{\boldsymbol{k}+V_{\overline{\boldsymbol{\mu}}}}),\ \boldsymbol{a} \preceq \overline{\boldsymbol{\mu}},\ \boldsymbol{k} \preceq \boldsymbol{\mu}. \tag{11}$$

When $\boldsymbol{\mu} = 0$ there is no subspace fixing, and Eq. (11) simplifies to the *periodic autocorrelation* of f, given by

$$p_{\boldsymbol{a}} = \mathcal{F}(\mathcal{D}_{\boldsymbol{a}} f). \tag{12}$$

Definition 2 ([1])**.** Let $E \subset \mathrm{GF}(2)^m$. The function $f \in \mathcal{B}_m$ satisfies the (periodic) *propagation criteria* (PC) with respect to E if, for all $\boldsymbol{e} \in E$, $p_{\boldsymbol{e}} = 0$. The function f satisfies PC of degree l and order q (also denoted PC(l) of order q) for some positive integers l and q if $p_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}} = 0$ for any $\boldsymbol{a}, \boldsymbol{k}, \boldsymbol{\mu} \in \mathrm{GF}(2)^m$ such that $\boldsymbol{a} \preceq \overline{\boldsymbol{\mu}}$, $\boldsymbol{k} \preceq \boldsymbol{\mu}$, $1 \leq \mathrm{wt}(\boldsymbol{a}) \leq l$, and $0 \leq \mathrm{wt}(\boldsymbol{\mu}) \leq q$. For $q = 0$ we abbreviate, saying that f satisfies PC(l).

3. Aperiodic Autocorrelation of a Boolean Function

For $\boldsymbol{a}, \boldsymbol{k}, \boldsymbol{\mu} \in \mathrm{GF}(2)^m$, $\boldsymbol{a}, \boldsymbol{k} \preceq \boldsymbol{\mu}$, and $\boldsymbol{\theta} = \boldsymbol{\mu} + \boldsymbol{a}$, where $\boldsymbol{\theta}$ and $\boldsymbol{a}$ are disjoint, the *fixed-aperiodic autocorrelation* coefficients of f after fixing the subspace $V_{\boldsymbol{\theta}}$ to $\boldsymbol{k} \,\&\, \boldsymbol{\theta}$ are defined by

$$u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}} = \mathcal{F}(\mathcal{D}_{\boldsymbol{a}} f \phi_{\boldsymbol{k}+V_{\overline{\boldsymbol{\mu}}}}),\ \boldsymbol{a}, \boldsymbol{k} \preceq \boldsymbol{\mu}. \tag{13}$$

The only difference between Eq. (11) and Eq. (13) is that, for the fixed-periodic case, $\boldsymbol{a} \preceq \overline{\boldsymbol{\mu}}$, whereas, for the fixed-aperiodic case, $\boldsymbol{a} \preceq \boldsymbol{\mu}$. For Eq. (11), $(\mathcal{D}_{\boldsymbol{a}} f)\phi_{\boldsymbol{k}+V_{\overline{\boldsymbol{\mu}}}} = \mathcal{D}_{\boldsymbol{a}}(f\phi_{\boldsymbol{k}+V_{\overline{\boldsymbol{\mu}}}})$, but this is ill-defined for Eq. (13). Note that "knowledge of the values of a fixed subset, $\boldsymbol{\mu}$", as stated in Section 1, is here characterised by fixed values of $\boldsymbol{k}$, where $\boldsymbol{k}$ is covered by $\boldsymbol{\mu}$.

When $\boldsymbol{\mu} = \boldsymbol{a}$ there are no additional fixed values, and Eq. (13) simplifies to the *aperiodic autocorrelation* of f, given by

$$u_{\boldsymbol{a},\boldsymbol{k}} = \mathcal{F}(\mathcal{D}_{\boldsymbol{a}} f \phi_{\boldsymbol{k}+V_{\overline{\boldsymbol{a}}}}),\ \boldsymbol{k} \preceq \boldsymbol{a}. \tag{14}$$

In other words, the aperiodic autocorrelation coefficients are given by a set of restrictions on the first derivatives of f. From the definitions there are $\sum_{\boldsymbol{a} \in \mathrm{GF}(2)^m} 2^{\mathrm{wt}(\boldsymbol{a})} = 3^m$ coefficients $u_{\boldsymbol{a},\boldsymbol{k}}$ and $\sum_{\boldsymbol{\mu} \in \mathrm{GF}(2)^m} 2^{2\,\mathrm{wt}(\boldsymbol{\mu})} = 5^m$ coefficients $u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}$. In fact, for autocorrelations of real functions,

$\mathrm{GF}(2)^m \to \mathbb{R}$, there are only a maximum of $\frac{3^m}{2}$ and $\frac{5^m}{2}$ different values for $u_{\boldsymbol{a},\boldsymbol{k}}$ and $u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}$, respectively.

The fixed-aperiodic autocorrelation of a Boolean function over a subspace is related to the *extended propagation criteria* (EPC) as defined by Preneel et al. [1], and investigated by Carlet [2]. However, the aperiodic property is more accurately characterised by a criteria we define as *aperiodic propagation criteria* (APC). We first explain why Eq. (13) is an aperiodic (nonmodular) metric, and we later return to the definitions of both APC and EPC.

Proposition 3. *The periodic autocorrelations of Eq.* (11) *and Eq.* (12) *can be expressed as modular (periodic) multivariate polynomial multiplications, and the aperiodic autocorrelations of Eq.* (13) *and Eq.* (14) *can be expressed as nonmodular (aperiodic) multivariate polynomial multiplications.*

Proof. Let $p_{\boldsymbol{a}}$ and $u_{\boldsymbol{a},\boldsymbol{k}}$ be as defined in Eq. (12) and Eq. (14). Let $\boldsymbol{z} \in \mathbb{C}^m$. Define $v(\boldsymbol{z})$, $P(\boldsymbol{z})$, and $A(\boldsymbol{z})$ as

$$v(\boldsymbol{z}) = \sum_{\boldsymbol{x} \in \mathrm{GF}(2)^m} (-1)^{f(\boldsymbol{x})} \prod_{i \in \mathbb{Z}_m} z_i^{x_i}, \tag{15}$$

$$P(\boldsymbol{z}) = \sum_{\boldsymbol{a} \in \mathrm{GF}(2)^m} p_{\boldsymbol{a}} \prod_{i \in \mathbb{Z}_m} z_i^{a_i}, \tag{16}$$

$$A(\boldsymbol{z}) = \sum_{\substack{\boldsymbol{k},\boldsymbol{a} \in \mathrm{GF}(2)^m \\ \boldsymbol{k} \preceq \boldsymbol{a}}} u_{\boldsymbol{a},\boldsymbol{k}} \prod_{i \in \mathbb{Z}_m} z_i^{a_i(-1)^{k_i}}. \tag{17}$$

Let $\boldsymbol{z}^{-1} = (z_0^{-1}, z_1^{-1}, \ldots, z_{m-1}^{-1})$. Then an expansion verifies the following modular and nonmodular relationships for $P(\boldsymbol{z})$ and $A(\boldsymbol{z})$.

$$P(\boldsymbol{z}) = v(\boldsymbol{z})v(\boldsymbol{z}^{-1}) \pmod{\prod_{i \in \mathbb{Z}_m} (z_i^2 - 1)}, \tag{18}$$

$$A(\boldsymbol{z}) = v(\boldsymbol{z})v(\boldsymbol{z}^{-1}). \tag{19}$$

The above argument carries over simply to Eq. (11) (resp. Eq. (13)) by first fixing a subspace $V_{\boldsymbol{\mu}}$ (resp. $V_{\boldsymbol{\theta}}$), then computing a modular (resp. nonmodular) polynomial multiplication over the remaining subspace. □

For $\boldsymbol{a}, \boldsymbol{c} \in \mathrm{GF}(2)^m$, define $G_{\boldsymbol{a},\boldsymbol{c}}$ as the Fourier spectrum of $\mathcal{D}_{\boldsymbol{a}} f$, so that

$$G_{\boldsymbol{a},\boldsymbol{c}} = \mathcal{F}(\mathcal{D}_{\boldsymbol{a}} f + \boldsymbol{c} \cdot \boldsymbol{x}). \tag{20}$$

The fixed-aperiodic autocorrelation of f after fixing a subspace, $V_{\boldsymbol{\theta}}$, is equivalent to a subspace Fourier transform of the Fourier transform of the first derivatives of f, as in the following proposition.

Proposition 4.

$$u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}} = 2^{-\operatorname{wt}(\boldsymbol{\mu})} \sum_{\boldsymbol{c} \preceq \boldsymbol{\mu}} G_{\boldsymbol{a},\boldsymbol{c}} (-1)^{\boldsymbol{k}\cdot\boldsymbol{c}},\ \boldsymbol{a},\boldsymbol{k} \preceq \boldsymbol{\mu}, \tag{21}$$

$$G_{\boldsymbol{a},\boldsymbol{c}} = \sum_{\boldsymbol{k} \preceq \boldsymbol{\mu}} u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}} (-1)^{\boldsymbol{c}\cdot\boldsymbol{k}},\ \boldsymbol{a},\boldsymbol{c} \preceq \boldsymbol{\mu}, \tag{22}$$

where, as before, the simplification to no additional fixed values is given by assigning $\boldsymbol{\mu} = \boldsymbol{a}$.

Proof. See Appendix A. □

The relationship between aperiodic autocorrelation and its constituent periodic and negaperiodic autocorrelations is described in subsection B.1 of Appendix B, and the relationships to the second derivative are described in subsection B.2 of the same appendix.

We can establish power relationships between fixed-aperiodic coefficients and Fourier spectra of the first derivative of f as follows.

$$\sum_{\boldsymbol{k} \preceq \boldsymbol{\mu}} |u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}|^2 = 2^{-\operatorname{wt}(\boldsymbol{\mu})} \sum_{\boldsymbol{c} \preceq \boldsymbol{\mu}} |G_{\boldsymbol{a},\boldsymbol{c}}|^2 \tag{23}$$

We define the *fixed-aperiodic sum-of-squares* with respect to $\boldsymbol{a}$ after fixing a subspace $V_{\boldsymbol{\theta}}$, referred to as $\sigma_{\boldsymbol{a},\boldsymbol{\mu}}$, as

$$\sigma_{\boldsymbol{a},\boldsymbol{\mu}} = \sum_{\boldsymbol{k} \preceq \boldsymbol{\mu}} |u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}|^2. \tag{24}$$

By summing over all $\boldsymbol{a},\boldsymbol{\mu} \in \mathrm{GF}(2)^m$ where $\boldsymbol{a} \preceq \boldsymbol{\mu}$, we arrive at an expression for the *complete fixed-aperiodic sum-of-squares*, $\mathcal{E}$, for f.

$$2\mathcal{E} + 6^n = \sum_{\boldsymbol{\mu} \in \mathrm{GF}(2)^m} \sum_{\boldsymbol{a} \preceq \boldsymbol{\mu}} \sigma_{\boldsymbol{a},\boldsymbol{\mu}} = \sum_{\boldsymbol{\mu} \in \mathrm{GF}(2)^m} \sum_{\boldsymbol{a},\boldsymbol{k} \preceq \boldsymbol{\mu}} |u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}|^2 \tag{25}$$

When $\boldsymbol{a} = \boldsymbol{\mu}$, the above expression simplifies to the *aperiodic sum-of-squares*, σ, where

$$2\sigma + 4^n = \sum_{\boldsymbol{a} \in \mathrm{GF}(2)^m} \sigma_{\boldsymbol{a}} = \sum_{\boldsymbol{a} \in \mathrm{GF}(2)^m} \sum_{\boldsymbol{k} \preceq \boldsymbol{a}} |u_{\boldsymbol{a},\boldsymbol{k}}|^2. \tag{26}$$

The aperiodic sum-of-squares, and the complete fixed-aperiodic sum-of-squares, have been investigated in [10] and [11], resp., where recursions in σ and $\mathcal{E}$, resp., have been established for certain infinite quadratic Boolean constructions.[1] Of significant interest in this paper are the choices for $\boldsymbol{a}$ and $\boldsymbol{\mu}$ such that $\sigma_{\boldsymbol{a},\boldsymbol{\mu}} = 0$, in particular for the cases where $\mathrm{wt}(\boldsymbol{\mu})$ is small. To this end we define the *aperiodic propagation criteria* as follows.

Definition 5. The function $f \in \mathcal{B}_m$ satisfies the *aperiodic propagation criteria* (APC) of degree l and order q (also denoted APC(l) of order q), for some positive integers l and q if $u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}} = 0$ for any $\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu} \in \mathrm{GF}(2)^m$ such that $\boldsymbol{a},\boldsymbol{k} \preceq \boldsymbol{\mu}$, $\boldsymbol{\mu} = \boldsymbol{a} + \boldsymbol{\theta}$, $1 \le \mathrm{wt}(\boldsymbol{a}) \le l$ and $0 \le \mathrm{wt}(\boldsymbol{\theta}) \le q$, where $\boldsymbol{a}$ and $\boldsymbol{\theta}$ are disjoint. For $q = 0$ we abbreviate, saying that f satisfies APC(l).

An intuitive reason for the usefulness of APC in a classical cryptographic context is as follows. Let $\boldsymbol{x} = \{x_i\}$ be the complete set of input bits. let $\boldsymbol{x}_{\boldsymbol{\mu}}, \boldsymbol{x}_{\boldsymbol{a}} \subseteq \boldsymbol{x}$ be such that $\boldsymbol{x}_{\boldsymbol{a}} \subseteq \boldsymbol{x}_{\boldsymbol{\mu}}$, $|\boldsymbol{x}_{\boldsymbol{\mu}}| \le q + |\boldsymbol{x}_{\boldsymbol{a}}|$, and $|\boldsymbol{x}_{\boldsymbol{a}}| \le l$. Then a Boolean function, f, satisfies APC(l) of order q if, for every possible $\boldsymbol{x}_{\boldsymbol{\mu}}, \boldsymbol{x}_{\boldsymbol{a}}$ pair, knowledge of the bits in $\boldsymbol{x}_{\boldsymbol{\mu}}$ gives no information as to the values of the function $\mathcal{D}_{\boldsymbol{a}} f$, where $a_i = 1$ iff $x_i \in \boldsymbol{x}_{\boldsymbol{a}}$. This definition is very similar but not identical to the *extended propagation criteria* (EPC) originally defined by Preneel et al. [1]. In order to define EPC, we first define *extended autocorrelation*.

For $\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu} \in \mathrm{GF}(2)^m$, $\boldsymbol{k} \preceq \boldsymbol{\mu}$, and $\boldsymbol{\theta} \preceq \boldsymbol{\mu}$, the *fixed-extended autocorrelation* coefficients of f after fixing the subspace, $V_{\boldsymbol{\theta}}$, to $\boldsymbol{k} \,\&\, \boldsymbol{\theta}$, are defined by

$$v_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}} = \mathcal{F}(\mathcal{D}_{\boldsymbol{a}} f \phi_{\boldsymbol{k}+V_{\bar{\boldsymbol{\mu}}}}),\ \boldsymbol{k} \preceq \boldsymbol{\mu}. \tag{27}$$

When $\boldsymbol{\mu} \preceq \boldsymbol{a}$, Eq. (27) simplifies to the *extended autocorrelation* of f, given by

$$v_{\boldsymbol{a},\boldsymbol{k}} = \mathcal{F}(\mathcal{D}_{\boldsymbol{a}} f \phi_{\boldsymbol{k}+V_{\bar{\boldsymbol{\mu}}}}),\ \boldsymbol{k} \preceq \boldsymbol{a}. \tag{28}$$

Note that

$$u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}} = v_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}},\ \boldsymbol{a} \preceq \boldsymbol{\mu}, \tag{29}$$

$$u_{\boldsymbol{a},\boldsymbol{k}} = v_{\boldsymbol{a},\boldsymbol{k}},\ \boldsymbol{a} = \boldsymbol{\mu}, \tag{30}$$

so the fixed-aperiodic autocorrelation coefficients are a subset of the extended autocorrelation coefficients. EPC is defined as follows.

[1] The factor of 2 on the left-hand sides of Eq. (25) and Eq. (26) reflects the fact that, for real functions, $\mathrm{GF}(2)^m \to \mathbb{R}$, we have $u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}} = u_{\boldsymbol{a},\bar{\boldsymbol{k}},\boldsymbol{\mu}}$ and $u_{\boldsymbol{a},\boldsymbol{k}} = u_{\boldsymbol{a},\bar{\boldsymbol{k}}}$, respectively. Moreover, 6^n and 4^n represent the zero-shift contributions.

Definition 6 ([1])**.** The function $f \in \mathcal{B}_m$ satisfies the *extended propagation criteria* (EPC) of degree l and order q (also denoted EPC(l) of order q) for some positive integers l and q if $v_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}} = 0$ for any $\boldsymbol{a}, \boldsymbol{k}, \boldsymbol{\mu} \in \mathrm{GF}(2)^m$, such that $\boldsymbol{k} \preceq \boldsymbol{\mu}$, $1 \leq \mathrm{wt}(\boldsymbol{a}) \leq l$ and $0 \leq \mathrm{wt}(\boldsymbol{\mu}) \leq q$. For $q = 0$ we abbreviate, saying that f satisfies EPC(l).[2]

An intuitive reason for the usefulness of EPC in a classical cryptographic context is as follows [1, 2]. Let $\boldsymbol{x} = \{x_i\}$ be the complete set of input bits. Let $\boldsymbol{x}_{\boldsymbol{\mu}}, \boldsymbol{x}_{\boldsymbol{a}} \subseteq \boldsymbol{x}$ be such that $|\boldsymbol{x}_{\boldsymbol{\mu}}| \leq q$, and $|\boldsymbol{x}_{\boldsymbol{a}}| \leq l$. Then a Boolean function, f, satisfies EPC(l) of order q if, for every possible $\boldsymbol{x}_{\boldsymbol{\mu}}, \boldsymbol{x}_{\boldsymbol{a}}$ pair, knowledge of the bits in $\boldsymbol{x}_{\boldsymbol{\mu}}$ gives no information as to the values of the function $\mathcal{D}_{\boldsymbol{a}} f$, where $a_i = 1$ iff $x_i \in \boldsymbol{x}_{\boldsymbol{a}}$.

The essential difference between APC and EPC is that, for APC the bits in the set $\boldsymbol{x}_{\boldsymbol{a}}$ are assumed to be known. This is not necessarily the case for EPC. In practice this means that APC envisages a scenario where the ability to modify input bits from the set $\boldsymbol{x}_{\boldsymbol{a}}$ also means that the attacker has "free" knowledge of the values of these same bits. In other words, "modify" and "read" are not distinguished for APC, whereas they are distinguished for EPC.

It is useful to define both APC and EPC in terms of one parameter each, namely *APC distance* and *EPC distance*.

Definition 7. The function $f \in \mathcal{B}_m$ has *APC distance* d if it satisfies APC(l) of order q for all positive integers, l, q, such that $d > l + q$.

Definition 8. The function $f \in \mathcal{B}_m$ has *EPC distance* d if it satisfies EPC(l) of order q for all positive integers, l, q, such that $d > l + q$.

The following is easily verified from Eq. (29).

$$\text{APC distance}(f) \leq \text{EPC distance}(f) \tag{31}$$

Computational results suggest that, for most Boolean functions of a small number of variables, the two distances are equal. A counterexample is the *clique function*, $f = \sum_{i<j} x_i x_j$. For $m \geq 4$, we have EPC distance $= 4$ but APC distance $= 2$.

The APC has been defined above in terms of fixed-aperiodic coefficients, $u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}$, but can also be defined in terms of $G_{\boldsymbol{a},\boldsymbol{c}}$. From Eq. (23) we have the following two-way implication, where $\boldsymbol{a} \preceq \boldsymbol{\mu}$.

$$u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}} = 0,\ \forall \boldsymbol{k} \preceq \boldsymbol{\mu} \quad \Leftrightarrow \quad G_{\boldsymbol{a},\boldsymbol{c}} = 0,\ \forall \boldsymbol{c} \preceq \boldsymbol{\mu}. \tag{32}$$

[2]There appears to be some disagreement in the literature regarding the distinction between PC and EPC, and the reader should be aware that some papers (e.g. [3]) refer to EPC(l) of order k as PC(l) of order k.

Preneel et al. [1] and Carlet [2] have given spectral characterisations of the EPC in terms of the Fourier transform of $\mathcal{D}_{\boldsymbol{a}} f$. We now re-express this characterisation in terms of the EPC distance and resilience of $\mathcal{D}_{\boldsymbol{a}} f$.

Corollary 9. *f has EPC distance d if and only if $\mathcal{D}_{\boldsymbol{a}} f$ is $(d - \mathrm{wt}(\boldsymbol{a}) - 1)$-resilient for all $\boldsymbol{a}$ where $\mathrm{wt}(\boldsymbol{a}) < d$.*

Using Eq. (31) we obtain the following corollary.

Corollary 10. *If f has APC distance d, then $\mathcal{D}_{\boldsymbol{a}} f$ is $(d - \mathrm{wt}(\boldsymbol{a}) - 1)$-resilient for all $\boldsymbol{a}$ where $\mathrm{wt}(\boldsymbol{a}) < d$.*

If $\mathcal{D}_{\boldsymbol{a}} f$ is $(d - \mathrm{wt}(\boldsymbol{a}) - 1)$-resilient, then f may have APC distance less than d, (e.g. the clique function $f = \sum_{i<j} x_i x_j$ for $m \geq 3$).

APC distance is slightly stricter than EPC distance[3] and both are much stricter criteria than PC. For example, it is easily verified that the *hyper-bent* function $f = x_0x_1x_2 + x_0x_1x_5 + x_0x_2x_3 + x_0x_4x_5 + x_1x_2x_3 + x_1x_2x_4 + x_1x_2x_5 + x_1x_3x_4 + x_1x_3x_5 + x_1x_4x_5 + x_2x_4x_5 + x_0x_3 + x_0x_5 + x_1x_4 + x_2x_3 + x_3x_4$ satisfies PC(6), but only APC(1), and further has both APC distance and EPC distance equal to 2. In fact, PC acts as an upper-bound on EPC which, in turn, acts as an upper bound on APC, giving the following lemma.

Lemma 11. *Let f satisfy* PC(l) *of order q,* EPC(l') *of order q, and* APC(l'') *of order q. Then $l'' \leq l' \leq l$.*

Fig. 1 shows the scope of $\boldsymbol{\mu}$ and $\boldsymbol{a}$ for EPC, APC, and PC. Although EPC is more general then APC (because $\boldsymbol{a}$ is not necessarily a subset of $\boldsymbol{\mu}$), the "spectral region" examined by EPC is no bigger than for APC. In other words, for EPC, the part of $\boldsymbol{a}$ not covered by $\boldsymbol{\mu}$ is, in a sense, superfluous, as it refers only to the periodic autocorrelation, which is a spectral subset of the aperiodic autocorrelation.[4] APC, on the other hand, has no purely periodic part.

[3] Although the fixed-aperiodic autocorrelation coefficients are a subset of the extended autocorrelation coefficients (see Eq. (29)), the interpretation of the weight of the coefficient indices as a distance measure means that APC is stricter than EPC. More informally, EPC distance is weaker than APC distance because EPC double-counts (does not identify) the overlap between $\boldsymbol{\mu}$ and $\boldsymbol{a}$.

[4] By "spectral region" we mean that the $u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}$ and $v_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}$ of f can both be computed from the $\{I, H, N\}^m$ set of transforms, where $\{I, H, N\}^m$ is as defined in Section 4.6. More specifically, aperiodic autocorrelation ($u_{\boldsymbol{a},\boldsymbol{k}}$) can be computed from the set of $\{H, N\}^m$ transform coefficients, whereas periodic autocorrelation ($p_{\boldsymbol{a}}$) can be computed from the $\{H\}^m$ (Walsh-Hadamard) coefficients, which are a subset of the $\{H, N\}^m$ transform coefficients.

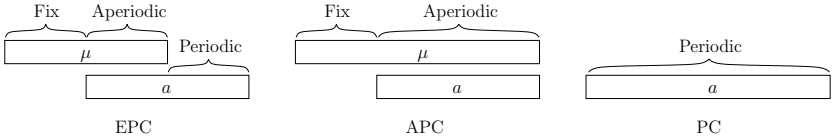


Fig. 1: *Relative Scope of $\boldsymbol{\mu}$ and $\boldsymbol{a}$ for Extended, Aperiodic, and Periodic Autocorrelations*

Here is a well-known quadratic construction [12] for $f \in \mathcal{B}_m$ which satisfies APC($\lfloor \frac{m}{2} \rfloor$).

Theorem 12. *Define $f \in \mathcal{B}_m$, $\boldsymbol{e} \in \mathrm{GF}(2)^m$, and $d \in \mathrm{GF}(2)$ such that*

$$f(\boldsymbol{x}) = \sum_{i=0}^{m-2} x_{\pi(i)} x_{\pi(i+1)} + \boldsymbol{e} \cdot \boldsymbol{x} + d, \tag{33}$$

where π is any permutation from $\mathbb{Z}_m$ to $\mathbb{Z}_m$. Then f satisfies APC($\lfloor \frac{m}{2} \rfloor$).

Proof. See Appendix A. □

Unfortunately the construction of Theorem 12 only gives APC distance 2. This is because fixing variables can comprise the strength of the residual subspace function. For instance, for π the identity, $\boldsymbol{\mu} = 1100\ldots$, and $\boldsymbol{a} = 100\ldots$ we find that $u_{\boldsymbol{a},k,\boldsymbol{\mu}} \neq 0$ and $\mathrm{wt}(\boldsymbol{\mu}) = 2$.

4. Constructions for Boolean Functions with Favourable APC

4.1. Qubits and Local Unitary Transforms

A *quantum bit* or *qubit* is an idealisation of a 2-dimensional quantum object. It is described by the vector (q_0, q_1), such that the probability of measuring the qubit in state 0 or state 1 is $|q_0|^2$ or $|q_1|^2$, respectively, with $|q_0|^2 + |q_1|^2 = 1$. Similarly, m qubits comprise a 2^m-dimensional object or *pure*[5] *quantum state*, $|\psi\rangle$, as described by the vector $\boldsymbol{s} = (s_{00\ldots0}, s_{00\ldots1}, \ldots, s_{11\ldots1})$ such that the probability of a joint measurement on the m qubits of $|\psi\rangle$ yielding state i is $|s_i|^2$, where $i \in \mathbb{Z}_2^m$, and $||\boldsymbol{s}||_2^2 = \sum_{i=00\ldots0}^{11\ldots1} |s_i|^2 = 1$, where $||\boldsymbol{s}||_p$ is the L_p-norm of $\boldsymbol{s}$. We say that $\boldsymbol{s}$ is normalised if $||\boldsymbol{s}||_2^2 = 1$. A local change of basis on the measurement axes is realised by evaluating $\boldsymbol{s}' = U\boldsymbol{s}$, where U is

[5] Only pure states are considered in this paper.

a $2^m \times 2^m$ *tensor-decomposable*, *unitary* matrix. U is unitary if $UU^\dagger = I$, where I is the identity and $\dagger$ means *transpose conjugate*. U is tensor-decomposable if it can be written as $U = U_0 \otimes U_1 \otimes \cdots \otimes U_{m-1}$, where the U_j are 2×2 unitary matrices. If U is of this form, then it is referred to as a *local unitary transform*. The transform is local because it is fully tensor-decomposed. We define $\boldsymbol{s}$ and $\boldsymbol{s}'$ to be *locally equivalent* if $\boldsymbol{s}' = U\boldsymbol{s}$ for U a local unitary transform. In such a case, $\boldsymbol{s}$ and $\boldsymbol{s}'$ are considered to be equivalent quantum states. It is this notion of equivalence that is exploited later in this section in the context of Boolean functions. As in [5], we will use a bijective mapping from a Boolean function, $f \in \mathcal{B}_m$, to a quantum state of m qubits, $|\psi\rangle$, as represented by $\boldsymbol{s}$.

$$|\psi\rangle \equiv \boldsymbol{s} = 2^{-\frac{m}{2}}(-1)^{f(\boldsymbol{x})}, \tag{34}$$

with $s_i = 2^{-\frac{m}{2}}(-1)^{f(i)}$. Consequently we refer to qubit i as x_i. This mapping allows us to view the fixed-aperiodic autocorrelation of a Boolean function in a quantum context. In particular we will see that the typical error model used to define a QECC can be related precisely to the operations associated with the fixed-aperiodic autocorrelation of a Boolean function. As the QECC error set is invariant to a local basis change, this means that, if $\boldsymbol{s} = 2^{-\frac{m}{2}}(-1)^{f(\boldsymbol{x})}$ and $\boldsymbol{s}' = 2^{-\frac{m}{2}}(-1)^{f'(\boldsymbol{x})}$ are locally equivalent, then f and f' have the same fixed-aperiodic autocorrelation profile.

4.2. Quantum Error Correcting Codes

Stabilizer QECCs [13] make excellent candidates for Boolean functions with favourable APC. An $[[m, k, d]]$ QECC is a code over m qubits of dimension k and minimum distance d, where each of the 2^k codewords can be thought of as a length 2^m normalised complex vector. The typical error-model for such a code assumes the occurrence of *no error*, *bit-flip*, *phase-flip*, or *combined phase-flip then bit-flip* error on each qubit independently. These errors are denoted I, X, Z, and Y. We introduce the *Pauli matrices*

$$\begin{aligned} I &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \\ Z &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = iXZ, \end{aligned} \tag{35}$$

where $i^2 = -1$. The Pauli matrices form a linear basis for all 2×2 complex unitary matrices. Let a quantum code of m qubits be subject

to an error, $\mathcal{E} = (\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_{m-1})$, such that $\mathcal{E}_j \in \{I, X, Z, Y\}$ acts on qubit j. An error from $\mathcal{E}$ can be described by the local unitary transform $U_{\mathcal{E}} = \mathcal{E}_0 \otimes \mathcal{E}_1 \otimes \cdots \otimes \mathcal{E}_{m-1}$, such that $\boldsymbol{s}' = U_{\mathcal{E}}\boldsymbol{s}$ takes $\boldsymbol{s}$ to the errored state $\boldsymbol{s}'$. The weight of the error vector is given by $\mathrm{wt}(\mathcal{E}) = |\{\mathcal{E}_j \mid \mathcal{E}_j \neq I\}|$, and an $[[m, k, d]]$ QECC can, by definition, detect any error vector of weight less than d.

It has been shown that any stabilizer QECC can be represented by a graph on m vertices [6–9, 14–17]. Quantum states with a graphical representation which have a direct interpretation as quadratic Boolean functions were also investigated in [5]. These turn out to be QECCs of dimension $k = 0$, and therefore correspond to the *graph states* recently proposed in [4, 9] as a consequence of the work of [18, 19]. These QECCs also correspond to *additive self-dual codes over* $\mathrm{GF}(4)$ [8, 20]. The mapping from an additive self-dual code over $\mathrm{GF}(4)$ to a graph can be understood by converting the generator matrix over $\mathrm{GF}(4)$ to an equivalent form, G, such that $G = \Gamma + \omega I$, where Γ is a symmetric $m \times m$ matrix over $\mathrm{GF}(2)$ with zero diagonal, and ω is a primitive element of $\mathrm{GF}(4)$. This conversion is always possible if the code is self-dual. Γ is then, simultaneously, the adjacency matrix for a simple graph that represents the graph state. In this paper we also interpret this graph state as a quadratic Boolean function

$$f(\boldsymbol{x}) = \sum_{j>i} \Gamma_{i,j} x_i x_j, \tag{36}$$

where the $\Gamma_{i,j}$ are entries of Γ. In other words, we exploit the equivalence of $[[m, 0, d]]$ stabilizer QECCs to quadratic Boolean functions via their interpretation as simple graphs. Conversely, we interpret a quadratic Boolean function as a graph state which, in turn, is a stabilizer QECC of dimension zero, using the mapping in Eq. (34). The QECC literature often refers to stabilizer states more abstractly as eigenvectors of a subset of error operators,[6] but, without loss of generality, we can associate these eigenvectors with specific states. When the dimension of the QECC is $k = 0$ the code coincides with a single quantum state which we interpret in this paper by a quadratic Boolean function and, if the minimum distance, d, of the code is high, the state is relatively robust to errors, implying that the state is highly *entangled* [4, 5]. Later in this

[6]The QECC is defined by finding a subset of error operators such that any codeword in the QECC is a joint eigenvector of all operators in the subset, i.e., the codeword is "stabilised" by this subset of error operators. The minimum distance of the QECC is then given by the minimum-weight error operator in the subset.

section we also use the mapping in Eq. (34) to find non-stabilizer QECCs via nonquadratic Boolean functions. A pure m-partite quantum state is unentangled if its associated state vector can be fully decomposed as a tensor product. Otherwise the quantum state is considered to be entangled. There are many metrics to describe the entanglement of an m-partite quantum state just as there are many metrics to describe the properties of an error-correcting code [5], (and, for large enough m, most of them are intractable to compute). For $m > 2$ any single metric is, inevitably, insufficient to describe the properties of the state or code. However, in this paper, we focus on the fixed-aperiodic properties of the state as giving a good indication of the entanglement of the state—certainly much more useful than just the periodic properties—with high APC distance indicating high entanglement.[7]

Let $|\psi\rangle$ be described by f, and $\boldsymbol{a} \in \mathrm{GF}(2)^m$ define the set of bit-flips $X_{\boldsymbol{a}}$, such that qubit x_j is bit-flipped if $j \in \{k \mid a_k = 1\}$. These bit-flips can also be described in terms of f,

$$|\psi\rangle \to X_{\boldsymbol{a}}(|\psi\rangle) \quad \Leftrightarrow \quad f(\boldsymbol{x}) \to f(\boldsymbol{x} + \boldsymbol{a}). \tag{37}$$

Similarly, for $\boldsymbol{c} \in \mathrm{GF}(2)^m$, the set of phase-flips $Z_{\boldsymbol{c}}$, where qubit x_j is phase-flipped if $j \in \{k \mid c_k = 1\}$, can be described in terms of f as

$$|\psi\rangle \to Z_{\boldsymbol{c}}(|\psi\rangle) \quad \Leftrightarrow \quad f(\boldsymbol{x}) \to f(\boldsymbol{x}) + \boldsymbol{c} \cdot \boldsymbol{x}. \tag{38}$$

Any combination of phase-flips followed by bit-flips on $|\psi\rangle$ can be described in terms of f as

$$|\psi\rangle \to X_{\boldsymbol{a}} Z_{\boldsymbol{c}}(|\psi\rangle) \quad \Leftrightarrow \quad f(\boldsymbol{x}) \to f(\boldsymbol{x} + \boldsymbol{a}) + \boldsymbol{c} \cdot \boldsymbol{x} + \boldsymbol{c} \cdot \boldsymbol{a}, \tag{39}$$

with a combined phase-flip then bit-flip occurring at the indices covered by $\boldsymbol{a} \,\&\, \boldsymbol{c}$. Note that $Z_{\boldsymbol{c}} X_{\boldsymbol{a}}(|\psi\rangle) = -X_{\boldsymbol{a}} Z_{\boldsymbol{c}}(|\psi\rangle)$, but to simplify the discussion in this paper we ignore post-multiplication by -1 and assume phase-flips are always performed before bit-flips.

The error-vector, $\mathcal{E}$, describing $X_{\boldsymbol{a}} Z_{\boldsymbol{c}}(|\psi\rangle)$, has weight $\mathrm{wt}(\boldsymbol{\mu})$, where $\boldsymbol{\mu} = \boldsymbol{a} + \overline{\boldsymbol{a}} \,\&\, \boldsymbol{c}$ (i.e. $\boldsymbol{\mu} = \boldsymbol{a}$ OR $\boldsymbol{c}$). To ensure that the QECC can detect all errors of weight less than d it is necessary and sufficient that, for

[7] In the physics literature there is an important subset of entanglement metrics, namely *entanglement monotones* [21]. We will not discuss these metrics in this paper but, instead, consider the weaker, more general notion of *entanglement criteria*. APC are certainly the latter but are also closely related to the former. The sum-of-squares metric, $\mathcal{E}$, of Eq. (25) will be shown in a future paper to be an entanglement monotone to within a trivial re-formulation.

$\mathrm{wt}(\boldsymbol{\mu}) < d$, all error states, $X_{\boldsymbol{a}}Z_{\boldsymbol{c}}(|\psi\rangle)$, are orthogonal to $|\psi\rangle$ with respect to the normal scalar product of vectors. If this is true then the QECC is an $[[m, 0, d]]$ code.

Theorem 13. *For $f \in \mathcal{B}_m$, let $|\psi\rangle$ be a $[[m, 0, d]]$ QECC, described by $\boldsymbol{s} = 2^{-\frac{m}{2}}(-1)^{f(\boldsymbol{x})}$. Then f has APC distance d. Conversely, if f has APC distance d, then $\boldsymbol{s}$ represents an $[[m, 0, d]]$ QECC, $|\psi\rangle$.*

Proof. See Appendix A. □

Remark. Theorem 13 holds for f of any algebraic degree, but when f has degree two we are considering stabilizer QECCs. In this case, the error-subset which forms the stabilizer can be identified with the subset of fixed-aperiodic (as opposed to periodic) propagations that identify all *linear structures* [22, 23].

In this paper we focus on QECCs of dimension zero as these relate to single Boolean functions. (Codes of higher dimension relate to sets of functions which will be dealt with in future work). An $[[m, 0, d]]$ QECC corresponds to an $(m, 2^m, d)$ self-dual additive code over $\mathrm{GF}(4)$. We distinguish between two types of self-dual additive code over $\mathrm{GF}(4)$. A code is of *Type II* if all codewords have even weight, otherwise it is of *Type I*. Bounds on the minimum distance of self-dual codes were given by Rains and Sloane [20]. Let d_I be the minimum distance of a Type I code of length m. Then d_I is upper-bounded by

$$d_I \leq \begin{cases} 2\left\lfloor \frac{m}{6} \right\rfloor + 1, & \text{if } m \equiv 0 \pmod 6 \\ 2\left\lfloor \frac{m}{6} \right\rfloor + 3, & \text{if } m \equiv 5 \pmod 6 \\ 2\left\lfloor \frac{m}{6} \right\rfloor + 2, & \text{otherwise.} \end{cases} \tag{40}$$

There is a similar bound on d_{II}, the minimum distance of a Type II code of length m,

$$d_{II} \leq 2\left\lfloor \frac{m}{6} \right\rfloor + 2. \tag{41}$$

A code that meets the appropriate bound is called *extremal*. These upper-bounds translate directly into upper-bounds on the APC distance for quadratic Boolean functions of m variables.

4.3. Spectral Equivalence and Local Complementation

Parker and Rijmen [5] observed that quantum states represented by the clique function, $f(\boldsymbol{x}) = \sum_{i<j} x_i x_j$, and the star function, $f(\boldsymbol{x}) =$

$\sum_{i=1}^{m-1} x_0 x_i$, are equivalent with respect to local unitary transforms (and further equivalent to the generalised GHZ (Greenberger-Horne-Zeilinger) state). It turns out that, for a special subset of local unitary transforms, for any pair of Boolean functions which are equivalent with respect to this transform set, the APC distance remains invariant. This invariance is already known in the context of QECCs, i.e., for quadratic Boolean functions, but the proof is extended to all Boolean functions in subsection 4.6, where the transform equivalence is described in more detail.[8]

We focus here on the quadratic equivalence which has been formulated as a graph symmetry by Glynn [7, 8], where the symmetry operation is referred to as *vertex-neighbour-complement* (VNC). It was also described independently by Hein et al. [4] and Van den Nest et al. [9]. In [15] this operation is explicitly described via repeated actions of the so-called $\{I, H, N\}^m$ transform set. The same operation also has a history in graph theory, where it is referred to as *local complementation* (LC) by Bouchet [6], who identified *isotropic systems* as being equivalent with respect to local complementation. LC also translates into the natural equivalence between self-dual additive codes over GF(4). Not surprisingly, isotropic systems and self-dual additive codes over GF(4) are very similar structures (if not identical). The LC symmetry rule can be described as follows.

Definition 14. If the quadratic monomial $x_i x_j$ occurs in the algebraic normal form of the quadratic Boolean function $f \in \mathcal{B}_m$, then x_i and x_j are mutual neighbours in the graph represented by f, as described by the $m \times m$ symmetric adjacency matrix Γ, where $\Gamma_{i,j} = \Gamma_{j,i} = 1$ iff $x_i x_j$ occurs in f, and $\Gamma_{i,j} = 0$ otherwise. For quadratic $f, f' \in \mathcal{B}_m$, f and f' are in the same *LC orbit* if

$$f'(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{\substack{j,k \in N_a \\ j \neq k}} x_j x_k \pmod 2, \tag{42}$$

where N_a comprises the neighbours of x_a in the graphical representation of f.

[8] Note, however, that Boolean functions of degree greater than two with APC distance d do not map to stabilizer QECCs as these functions no longer map to joint eigenvectors of the error-set. However, one can still interpret the functions as $[[m, 0, d]]$ QECCs, as all errored-states of error-weight less than d are orthogonal to the unerrored states and, for large d, the quantum state is highly entangled.

In the same way that a *bent function* f and its dual, $\tilde{f}$, are equivalent with respect to a Walsh-Hadamard transform [24], so the members of an LC-orbit represent flat spectra with respect to a certain set of local unitary transforms as described in subsection 4.6 [15]. Exploiting this generalised duality, one can show the following.

Theorem 15. *Let $f, f' \in \mathcal{B}_m$ such that f and f' are quadratic and in the same LC orbit. Then f and f' have the same APC distance.*

For example, the quadratic functions $f_h(\boldsymbol{x}) = x_0x_1 + x_0x_3 + x_0x_4 + x_1x_2 + x_1x_5 + x_2x_3 + x_2x_5 + x_3x_4 + x_4x_5$ and $f'_h(\boldsymbol{x}) = x_0(x_1 + x_2 + x_3 + x_4 + x_5) + x_1x_2 + x_2x_3 + x_3x_4 + x_4x_5 + x_5x_1$ are in the same orbit and therefore have the same APC distance (of 4). They are the two representations of the $[[6, 0, 4]]$ *Hexacode* up to graph isomorphism. The graphs associated with these two functions both have a maximum *independent set* of 2, but the maximum independent sets of the clique and star graph, which are two members of another LC orbit, are 1 and $m - 1$. In general, quadratic Boolean functions with high APC distance correspond to LC orbits that only comprise graphs with small maximum independent sets [25, 26].

To illustrate the interpretation of the graph as a self-dual additive code over GF(4), consider the Hexacode as represented by the Boolean function f_h defined above. According to Eq. (36), this function corresponds to the graph with adjacency matrix

$$\Gamma = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

A generator matrix for the $(6, 2^6, 4)$ additive code over GF(4) can then be written as

$$\Gamma + \omega I = \begin{pmatrix} \omega & 1 & 0 & 1 & 1 & 0 \\ 1 & \omega & 1 & 0 & 0 & 1 \\ 0 & 1 & \omega & 1 & 0 & 1 \\ 1 & 0 & 1 & \omega & 1 & 0 \\ 1 & 0 & 0 & 1 & \omega & 1 \\ 0 & 1 & 1 & 0 & 1 & \omega \end{pmatrix},$$

where ω is a primitive element in GF(4).

Table 1: *Number of LC Orbits of Graphs on m Vertices*

m	1	2	3	4	5	6	7	8	9	10	11	12
i_m	1	1	1	2	4	11	26	101	440	3132	40 457	1 274 068
t_m	1	2	3	6	11	26	59	182	675	3990	45 144	1 323 363

All self-dual additive codes over GF(4) of length m, i.e., the LC orbits of quadratic Boolean functions, have been classified, up to equivalence, by Calderbank et al. [20] for $m \leq 5$, by Höhn [27] for $m \leq 7$, by Hein et al. [4] for $m \leq 7$, by Glynn et al. [8] for $m \leq 9$, and by two of the authors of this paper [25, 28] for $m \leq 12$. The number of LC orbits up to isomorphism is given in Table 1, where i_m denotes the number of LC orbits of *connected* graphs on m vertices, and t_m denotes the total number of LC orbits. The values of i_m and t_m can also be found as sequences A090899 and A094927 in *The On-Line Encyclopedia of Integer Sequences* [29]. A database of orbit representatives up to $m = 12$ can be obtained from `http://www.ii.uib.no/~larsed/vncorbits/`.

4.4. Examples

Consider the following construction, known as the *quadratic residue construction*. Let p be a prime of the form $4k + 1$. Assign $a_{ij} = 1$ iff $j - i$ is a quadratic residue modulo p, and $a_{ij} = 0$ otherwise. (n is a quadratic residue modulo p iff there exists an m such that $m^2 \equiv n \pmod{p}$.) Let $f \in \mathcal{B}_p$ be a quadratic Boolean function defined by

$$f(\boldsymbol{x}) = \sum_{i<j} a_{ij} x_i x_j. \tag{43}$$

Then f has favourable APC distance. The $m \times m$ symmetric adjacency matrix Γ, where $\Gamma_{i,j} = \Gamma_{j,i} = 1$ iff $a_{i,j} = 1$, represents a *Paley graph* which is well-known in the graph-theoretic literature.

We extend the above construction by "bordering" the function. With f as defined above, let $g \in \mathcal{B}_{p+1}$ be a quadratic Boolean function defined by

$$g(\boldsymbol{x}) = f(\boldsymbol{x}) + x_p \sum_{i=0}^{p-1} x_i. \tag{44}$$

Then g has favourable APC distance.

As an example, for $p = 5$, $f(\boldsymbol{x}) = x_0x_1 + x_1x_2 + x_2x_3 + x_3x_4 + x_4x_0$, and $g(\boldsymbol{x}) = f(\boldsymbol{x}) + x_5(x_0 + x_1 + x_2 + x_3 + x_4)$. f has APC distance

3 and g has APC distance 4. The function g is unique over the 6-variable quadratics in achieving an optimal APC distance of 4, and corresponds to the unique $[[6, 0, 4]]$ QECC, known as the Hexacode. This function has been identified as being a highly entangled 6-qubit quantum state [5]. As another example, when $p = 29$, f has an APC distance of 11 and g has an APC distance of 12.

For $m = 12$ the QECC with optimal minimum distance is the *Dodecacode* which maps to a function with APC distance 6. Its LC orbit can be represented by the Boolean function $f(\boldsymbol{x}) = x_0x_3 + x_0x_7 + x_0x_8 + x_0x_9 + x_0x_{11} + x_1x_4 + x_1x_6 + x_1x_8 + x_1x_9 + x_1x_{10} + x_2x_5 + x_2x_6 + x_2x_7 + x_2x_{10} + x_2x_{11} + x_3x_6 + x_3x_8 + x_3x_{10} + x_3x_{11} + x_4x_6 + x_4x_7 + x_4x_9 + x_4x_{11} + x_5x_7 + x_5x_8 + x_5x_9 + x_5x_{10} + x_6x_9 + x_7x_{10} + x_8x_{11}$. It is interesting to note that both the Hexacode and Dodecacode can be represented by regular graphs with minimal vertex degree for every vertex, namely 3 and 5, these being one less than their respective minimum distances. These minimal representations appear to be possible for many optimal QECCs although not all [25]. In particular, a partial (but significant) search did not reveal a regular graph with vertex degree 11 in the LC orbit of the graph corresponding to the $[[30, 0, 12]]$ QECC. It remains an open problem as to whether a minimal representation exists for this graph.

We are also able to use the LC orbit to improve the resiliency of quadratic functions, combined with the addition of a suitable affine function. The addition of linear terms does not change the APC. The LC orbit is particularly useful in this context as the maximum resiliency achievable can change over the orbit. For example, as discussed previously, there are two representations of the Hexacode up to isomorphism, namely f_h and f'_h. One of these functions, f'_h, is bent, i.e. satisfies $\mathrm{PC}(n)$, and so cannot be resilient for any linear offset. The other function is correlation immune of order 1 and the maximum achievable resiliency is 0 by choosing, say, the balanced function, $f_h + x_0$. Typically the maximum achievable resiliency for functions with favourable APC will be low [22].

4.5. Aperiodic Properties of Nonquadratic Boolean Functions

To the best of our knowledge, QECCs represented by Boolean functions of degree greater than two have not been examined in the literature. These will, in general, be non-stabilizer QECCs, as the Boolean functions

no longer map to eigenvectors of the error set, so one must be careful how to use these QECCs. However APC remains well-defined for such functions. Cryptographically, we are particularly interested in Boolean functions of high degree so as to avoid potential algebraic attacks. From a quantum standpoint, in general, one may expect the QECC minimum distance to decrease as algebraic degree rises. We now consider the APC distance of such functions. These functions can also be referred to as *hypergraph states*. Note that Kurosawa and Satoh [3] and Carlet [2] have proposed nonquadratic Boolean functions with favourable EPC properties based on binary linear codes and binary Kerdock and Preparata nonlinear codes, respectively.

An exhaustive computer search [25], making use of the program *nauty* [30], reveals that no Boolean function of 4 or 5 variables and of degree greater than 2 has an APC distance greater than 2. However, there are 24 cubic functions of 6 variables which satisfy an APC distance of 3. These 24 functions are inequivalent with respect to the symmetries discussed in Appendix C. If we also consider the symmetry described in subsection 4.6, there are only 11 inequivalent such functions. For example, $f(\boldsymbol{x}) = x_1x_3x_5 + x_1x_2x_5 + x_3x_4x_5 + x_2x_4x_5 + x_0x_1x_3 + x_0x_1x_2 + x_0x_3x_4 + x_0x_2x_4 + x_0x_4 + x_0x_5 + x_1x_2 + x_1x_4 + x_2x_3 + x_2x_5 + x_3x_4 + x_3x_5 + x_4x_5$ has APC and EPC distances of 3. It was also found that no cubic functions of 6 variables can achieve an APC distance greater than 3. By searching all inequivalent Boolean functions with just one nonquadratic term we found 7-variable and 8-variable functions with APC distances 3 and 4, respectively. For example, $f(\boldsymbol{x}) = x_1x_3x_5 + x_0x_1 + x_0x_2 + x_1x_6 + x_2x_5 + x_3x_4 + x_3x_6 + x_4x_5 + x_5x_6$ and $f(\boldsymbol{x}) = x_0x_1x_2x_3 + x_0x_4 + x_0x_5 + x_1x_4 + x_1x_6 + x_2x_5 + x_2x_6 + x_3x_4 + x_3x_5 + x_3x_6$ have APC and EPC distances of 3, and $f = x_0x_1x_2 + x_0x_4 + x_0x_5 + x_0x_7 + x_1x_4 + x_1x_6 + x_1x_7 + x_2x_5 + x_2x_6 + x_2x_7 + x_3x_4 + x_3x_5 + x_3x_6$ and $f = x_0x_1x_2x_3 + x_0x_4 + x_0x_5 + x_0x_6 + x_1x_4 + x_1x_5 + x_1x_7 + x_2x_4 + x_2x_6 + x_2x_7 + x_3x_5 + x_3x_6 + x_3x_7$ have APC and EPC distances of 4. These results equal the best distances achievable using quadratic functions.

The *Maiorana-McFarland construction* [24] is as follows.

$$f(\boldsymbol{y}, \boldsymbol{z}) = \boldsymbol{y} \cdot \lambda(\boldsymbol{z}) + g(\boldsymbol{z}), \tag{45}$$

where $f \in \mathcal{B}_{r+s}$, $\boldsymbol{y} \in \mathrm{GF}(2)^r$, $\boldsymbol{z} \in \mathrm{GF}(2)^s$, $g \in \mathcal{B}_s$, and λ maps $\mathrm{GF}(2)^s$ to $\mathrm{GF}(2)^r$. Following [3], the above examples of 7-variable and 8-variable functions can both be described using Eq. (45) with λ a linear map and $g(\boldsymbol{z})$ the nonquadratic part. We have found, as shown

above, functions of this kind with favourable APC but, as pointed out by Carlet [2], the reliance on $g(z)$ to make the function nonquadratic may lead to cryptanalytic attacks. A more interesting set of functions is obtained by changing λ to a nonlinear mapping. Carlet constructs such functions with favourable EPC [2], based on nonlinear Kerdock/Preparata mappings. We can, trivially, use Lemma 11 to state that, for these Kerdock/Preparata-based constructions, the resultant 2^{m+1}-variable functions satisfy APC(l) of order $2^{m-1} - 2^{m/2-1} - 1$, with maximum possible $l \leq 5$, or APC(l) of order 5 with maximum possible $l \leq 2^{m-1} - 2^{m/2-1} - 1$. Moreover, using Eq. (31), both the EPC and APC distances for such functions are upper-bounded by $2^{m-1} - 2^{m/2-1} + 5$. From Eq. (45), the Maiorana-McFarland construction is bipartite, and the size of the maximum independent set of its associated hypergraph is at least r. Typically one chooses $r = s$, but LC orbits of the graphs corresponding to the best QECCs maintain a small maximum independent set for every member of the orbit, i.e., $r \ll s$, with $g(z)$ an APC-favourable sub-graph. We expect, similarly, that constructions for Boolean functions of algebraic degree greater than two (hypergraphs) with favourable APC should also have a small independent set for their quadratic part, with $g(z)$ constructed recursively in the same way. Over 32 variables, the Maiorana-McFarland constructions of Carlet [2] satisfy an APC distance upper-bounded by 11 and the maximum independent set of the quadratic part of the functions is 16. In contrast the 30-variable function of subsection 4.4 has APC distance 12, and the graph describing this quadratic function has a maximum independent set of only 6. Moreover a partial search of about 10 million functions from within the (huge) LC orbit of this 30-variable function did not reveal a maximum independent set of size greater than 7.

4.6. Orbits of Boolean Functions with Respect to $\{I, H, N\}^m$

We describe how an orbit of Boolean functions can be generated such that any two members of the orbit are spectral "duals" with respect to a certain local unitary transform taken from a set of transforms called the $\{I, H, N\}^m$ set (using and refining the terminology introduced in [31]). The APC distance is invariant over this orbit.

For $\boldsymbol{a}, \boldsymbol{b} \in \mathrm{GF}(2)^m$, we define $\boldsymbol{a} \tilde{+} \boldsymbol{b}$ such that $0 \tilde{+} 0 = 0$, $1 \tilde{+} 0 = 0 \tilde{+} 1 = 1$, and $1 \tilde{+} 1 = 2$. Moreover, for $h \in \mathrm{GF}(2)$ and $c \in \mathbb{Z}$, we define ch to be in $\{0, c\}$.

Let $f \in \mathcal{B}_m$ and $\boldsymbol{\theta}, \boldsymbol{r}, \boldsymbol{\alpha}, \boldsymbol{e} \in \mathrm{GF}(2)^m$ such that $\boldsymbol{r} \preceq \boldsymbol{\theta}$ and $\boldsymbol{\alpha}, \boldsymbol{e} \preceq \overline{\boldsymbol{\theta}}$. Then each pair of values of $\boldsymbol{e}$ and $\boldsymbol{\theta}$ describes one of 3^m possible local unitary transforms taken from the $\{I, H, N\}^m$ set.

$$s_{\boldsymbol{e},\boldsymbol{\theta}}(\boldsymbol{z}) = 2^{\frac{\mathrm{wt}(\boldsymbol{\theta})}{2}} \sum_{\boldsymbol{x} \in \boldsymbol{r} + V_{\overline{\boldsymbol{\theta}}}} i^{2(f(\boldsymbol{x}) + \boldsymbol{\alpha}) \dotplus \boldsymbol{e}}, \tag{46}$$

where $\boldsymbol{z} = \boldsymbol{\alpha} + \boldsymbol{r}$, $i^2 = -1$, and $s_{\boldsymbol{e},\boldsymbol{\theta}} \in \mathbb{C}^{2^m}$. In related papers [5, 15, 31] the $\{I, H, N\}^m$ transform set is described as the set of 3^m local unitary transform matrices of size $2^m \times 2^m$, constructed from any possible tensor product combination of the 2×2 unitary matrices I, H, and N, defined as

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad N = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix}, \tag{47}$$

where $i^2 = -1$. In this paper we largely avoid the matrix terminology but retain the name $\{I, H, N\}^m$.[9]

If, for a fixed $\boldsymbol{e}$ and $\boldsymbol{\theta}$, $s_{\boldsymbol{e},\boldsymbol{\theta}}$ is a flat spectrum, i.e., if $|s_{\boldsymbol{e},\boldsymbol{\theta}}(\boldsymbol{z})| = |s_{\boldsymbol{e},\boldsymbol{\theta}}(\boldsymbol{z}')|$ for all $\boldsymbol{z}, \boldsymbol{z}' \in \mathrm{GF}(2)^m$, then we can write

$$s_{\boldsymbol{e},\boldsymbol{\theta}}(\boldsymbol{z}) = 2^{\frac{m}{2}} w^{g_{\boldsymbol{e},\boldsymbol{\theta}}(\boldsymbol{z})}, \tag{48}$$

where $g_{\boldsymbol{e},\boldsymbol{\theta}}(\boldsymbol{z})$ is a function from $\mathrm{GF}(2)^m$ to $\mathbb{Z}_8^m$ and $w = e^{\frac{2\pi i}{8}}$, $w \in \mathbb{C}$.

Definition 16. Let $f, f' \in \mathcal{B}_m$. Then f and f' are in the same $\{I, H, N\}^m$ orbit iff, for some choice of $\boldsymbol{e}$ and $\boldsymbol{\theta}$, $s_{\boldsymbol{e},\boldsymbol{\theta}}$ is a flat spectrum and $g_{\boldsymbol{e},\boldsymbol{\theta}}$ can further be written as $g_{\boldsymbol{e},\boldsymbol{\theta}}(\boldsymbol{z}) = 4f'(\boldsymbol{z}) + \boldsymbol{c} \cdot \boldsymbol{z} + d \pmod 8$, where $\boldsymbol{c} \in \mathbb{Z}_8^m$, and $d \in \mathbb{Z}_8$.

The following theorem has previously been proven for f quadratic but not for general f, which is proven here. The LC symmetry discussed in subsection 4.3 is a translation of the quadratic case of this theorem into graphical operations.

Theorem 17. *Let $f, f' \in \mathcal{B}_m$. If f and f' are both in the same $\{I, H, N\}^m$ orbit, then f and f' have the same APC distance.*

Proof. The proof relies on two critical observations that we express as lemmas.

[9] However, to clarify Eq. (46) in terms of $\{I, H, N\}^m$, note that the one positions in $\boldsymbol{\theta}$ and $\boldsymbol{e}$ identify the tensor positions where I and N are applied, respectively, with H applied to all other tensor positions.

Lemma 18. *Let $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{C}^N$ be two complex vectors of length N. Let U be an $N \times N$ complex unitary matrix such that $\boldsymbol{a}' = U\boldsymbol{a}$ and $\boldsymbol{b}' = U\boldsymbol{b}$. Define orthogonality of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ with respect to the scalar product, $\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \boldsymbol{a} \cdot \boldsymbol{b} = 0$. Then $\langle \boldsymbol{a}, \boldsymbol{b} \rangle = 0$ if and only if $\langle \boldsymbol{a}', \boldsymbol{b}' \rangle = 0$.*

Let $\mathcal{E} \in \{I, X, Y, Z\}$, as defined in Section 4, be the error acting on a single qubit. Then it can be shown that any transform, T, taken from the $\{I, H, N\}$ set for $m = 1$, takes the error set, $\{I, X, Y, Z\}$ to itself under conjugation. This is because the $\{I, H, N\}$ set generates the *local Clifford group* which is defined as the group of local unitary matrices that keeps the Pauli matrices over a single complex variable invariant with respect to conjugation [32] (to within a global constant). Explicitly, for $T \in \{I, H, N\}$, $\mathcal{E}' = T\mathcal{E}T^{-1}$ satisfies, $\mathcal{E}' \in \{I, X, Y, Z\}$.[10] It follows immediately that the $\{I, H, N\}^m$ transform set, as defined in Eq. (46), keeps $\mathcal{E}$ within the Pauli set for any fixed m, and keeps the weight of $\mathcal{E}$ invariant. We then arrive at the following lemma.

Lemma 19. *Let $T_{e,\theta} \in \{I, H, N\}^m$ and $\mathcal{E} \in \{I, X, Y, Z\}^m$. Then*

$$\mathcal{E}' = T_{e,\theta}\mathcal{E}T_{e,\theta}^{-1} \quad \Rightarrow \quad \mathcal{E}' \in \{I, X, Y, Z\}^m \quad \Rightarrow \quad \mathrm{wt}(\mathcal{E}') = \mathrm{wt}(\mathcal{E}). \tag{49}$$

Let a quantum state of m qubits, $|\psi\rangle$, be represented by a length 2^m vector $\boldsymbol{s} \in \mathbb{C}^{2^m}$, where $\boldsymbol{s} = 2^{-\frac{m}{2}}(-1)^{f(\boldsymbol{x})}$. We can then re-express Theorem 13 as follows.

$$\text{APC distance}(f) = d \quad \Rightarrow \quad \langle \mathcal{E}\boldsymbol{s}, \boldsymbol{s} \rangle = 0, \ \forall \mathcal{E}, \ 0 < \mathrm{wt}(\mathcal{E}) < d, \tag{50}$$

where $\mathcal{E} \in \{I, X, Y, Z\}^m$. We wish to show that

$$\text{APC distance}(f) = d \quad \Rightarrow \quad \langle \mathcal{E}'\boldsymbol{s}', \boldsymbol{s}' \rangle = 0, \ \forall \mathcal{E}', \ 0 < \mathrm{wt}(\mathcal{E}') < d, \tag{51}$$

where $\mathcal{E}' \in \{I, X, Y, Z\}^m$, and $\boldsymbol{s}'$ is any vector that occurs as a spectral output with respect to any transform taken from the $\{I, H, N\}^m$ set. To do this we note that $\boldsymbol{s} = T_{e,\theta}\boldsymbol{s}'$ for some $T_{e,\theta} \in \{I, H, N\}^m$. We now use Lemma 19 to conjugate $\mathcal{E}$ acting on $\boldsymbol{s}$ to $\mathcal{E}'$ acting on $\boldsymbol{s}'$. Now we can write $\langle \mathcal{E}\boldsymbol{s}, \boldsymbol{s} \rangle = 0$ as $\left\langle T_{e,\theta}^{-1}\mathcal{E}'T_{e,\theta}\boldsymbol{s}, T_{e,\theta}^{-1}T_{e,\theta}\boldsymbol{s} \right\rangle = 0$. It follows from Lemmas 18 and 19 that $\langle \mathcal{E}'T_{e,\theta}\boldsymbol{s}, T_{e,\theta}\boldsymbol{s} \rangle = 0, \ \forall \mathcal{E}', \ 0 < \mathrm{wt}(\mathcal{E}') < d$. The theorem follows. □

[10] Note that conjugation by H takes X to Z, Z to X, and Y to $-Y$. Conjugation by N takes X to $-iY$, Z to X, and Y to $-Z$. Conjugation by I takes X to X, Z to Z, and Y to Y.

Remark. Note that we have proved the invariance of the APC distance for any $\boldsymbol{s}$ and $\boldsymbol{s}'$ in the same orbit with respect to the $\{I, H, N\}^m$ transform set. So the proof not only holds for Boolean functions, but also more generally for functions from $\mathrm{GF}(2)^m$ to $\mathbb{Z}_8$. More generally still, the proof holds for any $\boldsymbol{s}$ and $\boldsymbol{s}'$, even when $\boldsymbol{s}$ and $\boldsymbol{s}'$ represent non-flat spectra.

We next provide an example of this spectral symmetry for nonquadratic Boolean functions, which generalises LC and uses the flat spectra of a Boolean function with respect to the $\{I, H, N\}^n$ transform set to generate an orbit of Boolean functions with the same APC distance, as described above. Consider the cubic Boolean function $x_0x_1x_2 + x_0x_1x_3 + x_0x_1x_4 + x_0x_2x_3 + x_0x_2x_4 + x_0x_5 + x_1x_3 + x_1x_5 + x_2x_4 + x_2x_5 + x_3x_4$ which has APC distance 3. Applying the transform technique described above, we obtain 144 flat spectra of which 20 map to Boolean functions. Of these 20, only 3 are inequivalent. These 3 functions are cubic and have APC distance 3 and EPC distance 3. For instance, $x_0x_1x_5 + x_0x_3x_5 + x_0x_4x_5 + x_0x_1 + x_0x_2 + x_0x_3 + x_0x_5 + x_1x_2 + x_1x_3 + x_1x_4 + x_1x_5 + x_2x_5 + x_3x_4 + x_4x_5$ is in the same orbit and is obtained via the transform obtained by setting $\boldsymbol{\theta} = 110110$ and $\boldsymbol{e} = 001000$. Note, however, that no linear offset of a member of this orbit is balanced, so resiliency cannot be satisfied.

5. Conclusions

We have motivated and characterised aperiodic autocorrelation and the aperiodic propagation criteria (APC) for a Boolean function. In particular we have equated, for quadratic Boolean functions, APC distance with the minimum distance of an associated zero-dimensional quantum error-correcting code. It follows that, for quantum states which have an interpretation as Boolean functions, the APC of the function are also quantum entanglement criteria for the associated state. We highlighted the importance of local complementation (LC) symmetry for APC analysis of quadratic Boolean functions, and also gave a generalisation of LC to Boolean functions of algebraic degree greater than two. We presented some results for the APC distance of functions of degree greater than two and discussed possible forms other Boolean constructions might take to improve APC distance.

We also showed that fixed-aperiodic autocorrelation is a subset of extended autocorrelation. We further defined the metrics of APC

and EPC distance and demonstrated that APC distance is a slightly stricter criteria than EPC distance. Although extended autocorrelation considers a slightly more general set of cryptographic scenarios than fixed-aperiodic autocorrelation, the APC, in some sense, highlights the most important parts of EPC, and this motivates the use of APC for cryptography.

APC is also a potential attack scenario. Just as generalised linear cryptanalysis [31] finds substantially higher biases over state-of-the-art S-boxes, the differential "dual", as covered in this paper, finds substantially higher differential biases where, by "differential" we here refer to an input differential $\Delta \boldsymbol{x} \in \mathrm{GF}(2)^m$, and an output binary (truncated) differential $\Delta y \in \mathrm{GF}(2)$. Appendix D gives results of an exhaustive search for the worst-case differential biases of given input differential weight, taken over the linear space of selected state-of-the-art S-boxes. It is evident that significantly higher biases can be obtained by using aperiodic as opposed to periodic differentials. One should remember that the context in which the S-box is used will determine whether a high-bias differential constitutes a weakness for the cipher. For instance, the 9×9 Misty1 S-box, because it is a quadratic S-box, has a linear space with periodic differential biases that occur with probability 1 for all weights, (i.e. it has linear structures for all weights), but these do not necessarily constitute a weakness as the S-box is used in a Feistel structure, and in conjunction with a 7×7 cubic S-box.[11] Still, the 7×7 S-box exhibits significantly higher aperiodic and fixed-aperiodic biases compared to periodic biases. These biases may lead to a practical block cipher attack. However, for the typical block cipher which inputs the key via XOR, one cannot exploit these higher biases by using the standard technique of piecing together differential trails through successive cipher rounds, as the "route" of the trail will be key-dependent [31, 34]. In other words, although aperiodic and fixed-aperiodic differentials establish much higher biases across constituent S-boxes and, by implication, across complete block ciphers, than periodic differentials, the location of these biases across multiple rounds is strongly key-dependent. So it may be difficult to exploit these high biases. Even so, the results of this paper provide an extended theoretical framework for a Boolean function, which suggests a technique where one finds a function with favourable fixed-aperiodic criteria, then one traverses, either exactly or approximately, through the

[11] However, see [33].

orbit generated by a set of local unitary transforms, so as to optimise the function with respect to the Walsh-Hadamard spectral criteria.

The problem of designing an S-box (or block cipher) so that all constituent Boolean functions have high APC distance is also an interesting challenge, but the stipulation that an S-box is a balanced function from $\mathrm{GF}(2)^m$ to $\mathrm{GF}(2)^n$ may limit the achievable APC distance. Note that all S-boxes examined in Appendix D achieve only APC distance 1 over the complete linear space of the S-box. (In fact most S-boxes are not even designed to achieve $\mathrm{PC}(1)$.) At the end of Table 3 we have included the worst-case biases for the single quadratic Boolean function that represents the $[[6, 0, 4]]$ Hexacode. By definition, the biases are all 0.5 up to weight 4. However it is much more constraining—and remains an open problem—to construct a function (S-box) with output in $\mathrm{GF}(2)^n$, $n > 1$, such that the low-weight biases of the linear space of the S-box are all near to 0.5. Finally, functions with favourable APC distance automatically have high generalised nonlinearity with respect to the generalised transform sets discussed by [31] and [15], e.g., with respect to $\{I, H, N\}^m$. This can be explained by considering a generalisation of the results of [35] to larger transform sets.

ACKNOWLEDGEMENTS The authors would like to thank Prof. Alexander Pott for reading early versions of this paper and for helpful suggestions, and Prof. Patrick Solé for helpful advice and for pointing out numerous connections with other work in the literature.

A. PROOFS

Proposition 4. Proposition 1 of [36] states

$$\sum_{\boldsymbol{v}\in V^\perp} \mathcal{F}(f + \boldsymbol{x}\cdot\boldsymbol{v}) = 2^{m-k}\mathcal{F}(f\phi_V), \tag{52}$$

where k is the dimension of V. Applying Eq. (52) to Eq. (20) gives

$$\sum_{\boldsymbol{c}\preceq\boldsymbol{\mu}} G_{\boldsymbol{a},\boldsymbol{c}} = \sum_{\boldsymbol{c}\preceq\boldsymbol{\mu}} \mathcal{F}(\mathcal{D}_{\boldsymbol{a}} f + \boldsymbol{c}\cdot\boldsymbol{x}) = 2^{\mathrm{wt}(\boldsymbol{\mu})}\mathcal{F}(\mathcal{D}_{\boldsymbol{a}} f \phi_{V_{\overline{\boldsymbol{\mu}}}}). \tag{53}$$

It is further stated in [36] that

$$\sum_{\boldsymbol{v}\in V^\perp} \mathcal{F}(f + \boldsymbol{x}\cdot\boldsymbol{v})(-1)^{\boldsymbol{k}\cdot\boldsymbol{v}} = 2^{m-k}\mathcal{F}(f\phi_{\boldsymbol{k}+V}). \tag{54}$$

Applying Eq. (54) to Eq. (13), Eq. (20), and Eq. (53) gives the result. □

Theorem 12. First we compute the values of $u_{\boldsymbol{a},\boldsymbol{k}}$ for $\boldsymbol{k} = \boldsymbol{0} = 000\ldots$ with π the identity permutation. Let $u_{\boldsymbol{a},\boldsymbol{k}}[m]$ denote the values of $u_{\boldsymbol{a},\boldsymbol{k}}$ for f over m variables. Below are tabulated the values of $u_{\boldsymbol{a},\boldsymbol{0}}[m]$ and the associated upper bound on the l of APC(l) inferred from these $u_{\boldsymbol{a},\boldsymbol{0}}[m]$, for all possible assignments to the three least significant bits (lsbs) of $\boldsymbol{a}$, where * means "don't care".

a (lsbs on the left)	$u_{\boldsymbol{a},\boldsymbol{0}}[m]$	Upper bound on l
$100\ldots$	0	m
$01^*\ldots$	0	m
$11^*\ldots$	$u_{\boldsymbol{a},\boldsymbol{0}}[m-1]$	$(m-1)+1=m$
$001\ldots$	0	m
$101\ldots$	$u_{\boldsymbol{a},\boldsymbol{0}}[m-2]$	$(m-2)+1=m-1$

We are interested in the lowest value of l that we can achieve by suitable assignments to $\boldsymbol{a}$. From the above table, the only case where the upper bound on l is lower than m is in the last row of the table. We recursively assign the lsbs of $\boldsymbol{a}$ according to this last row, (e.g., for the second iteration we have $\boldsymbol{a} = 10101\ldots$ and $l \leq m-2$). By induction one concludes that $l = \lfloor \frac{m}{2} \rfloor$. As f is a quadratic function we can invoke the symmetry of Lemma 21 in Appendix C to extend the result from $u_{\boldsymbol{a},\boldsymbol{0}}[m]$ to all $u_{\boldsymbol{a},\boldsymbol{k}}[m]$. We further invoke the permutation symmetry of Lemma 22 to extend the result to all functions f where π is not necessarily the identity permutation. □

Theorem 13. Consider all bit-flip and phase-flip errors on $|\psi\rangle$ of weight less than d, described by $\boldsymbol{a}$ and $\boldsymbol{c}$ such that $\mathrm{wt}(\boldsymbol{\mu}) = \mathrm{wt}(\boldsymbol{a}) + \mathrm{wt}(\boldsymbol{\theta}) < d$, as discussed previously, where $\boldsymbol{\mu} = \boldsymbol{a} + \overline{\boldsymbol{a}} \,\&\, \boldsymbol{c}$ and $\boldsymbol{\theta} = \overline{\boldsymbol{a}} \,\&\, \boldsymbol{c}$. We know that $X_{\boldsymbol{a}} Z_{\boldsymbol{c}} |\psi\rangle$, is orthogonal to $|\psi\rangle$ and this can be interpreted in terms of f by asserting that $\mathcal{D}_{\boldsymbol{a}} f + \boldsymbol{c} \cdot \boldsymbol{x}$ is balanced for all $\boldsymbol{a}, \boldsymbol{c}$ that satisfy $\mathrm{wt}(\boldsymbol{\mu}) < d$. In other words, from Eq. (20), Eq. (32), and Definition 6, $G_{\boldsymbol{a},\boldsymbol{c}} = 0$ for all $\boldsymbol{a}, \boldsymbol{c} \preceq \boldsymbol{\mu}$. The first part of the theorem follows from Definition 7. The converse is easily proven. □

B. Further Spectral Identities

B.1. Periodic/Negaperiodic Autocorrelation

We here define the *periodic/negaperiodic autocorrelation* of f, and show how its coefficients are derived from the Fourier spectra of $\mathcal{D}_{\boldsymbol{a}} f$, thus allowing us to relate the periodic/negaperiodic autocorrelation with the

aperiodic autocorrelation. The reason we refer to the autocorrelations as "periodic/negaperiodic" will be explained in Proposition 20. Define the *periodic/negaperiodic autocorrelation coefficients* of f after fixing the subspace $V_{\boldsymbol{\theta}}$ as $U_{\boldsymbol{a},\boldsymbol{e},\boldsymbol{r},\boldsymbol{\mu}}$, where $\boldsymbol{a},\boldsymbol{r},\boldsymbol{\mu} \in \mathrm{GF}(2)^m$, $\boldsymbol{e} \preceq \boldsymbol{a} \preceq \boldsymbol{\mu}$, $\boldsymbol{r} \preceq \boldsymbol{\theta}$, and $\boldsymbol{\theta} = \boldsymbol{\mu} + \boldsymbol{a}$, and $\boldsymbol{\theta}$ and $\boldsymbol{a}$ are disjoint. Then

$$\begin{aligned} U_{\boldsymbol{a},\boldsymbol{e},\boldsymbol{r},\boldsymbol{\mu}} &= 2^{-\mathrm{wt}(\boldsymbol{\theta})} \sum_{\boldsymbol{c} \in \boldsymbol{e}+V_{\boldsymbol{\theta}}} \mathcal{F}(\mathcal{D}_{\boldsymbol{a}} f + \boldsymbol{c}\cdot\boldsymbol{x} + \mathrm{wt}(\boldsymbol{c}))(-1)^{\boldsymbol{r}\cdot\boldsymbol{c}} \\ &= 2^{-\mathrm{wt}(\boldsymbol{\theta})} \sum_{\boldsymbol{c} \in \boldsymbol{e}+V_{\boldsymbol{\theta}}} \mathcal{F}(\mathcal{D}_{\boldsymbol{a}} f + \boldsymbol{c}\cdot\boldsymbol{x})(-1)^{\overline{\boldsymbol{r}}\cdot\boldsymbol{c}}. \end{aligned} \tag{55}$$

When $\boldsymbol{\mu} = \boldsymbol{a}$ then $\boldsymbol{\theta} = \boldsymbol{0}$ and there is no subspace fixing, so that Eq. (55) simplifies to the computation of the periodic/negaperiodic autocorrelation coefficients of f, namely $U_{\boldsymbol{a},\boldsymbol{c}}$, where $\boldsymbol{c} \preceq \boldsymbol{a}$.

$$U_{\boldsymbol{a},\boldsymbol{c}} = (-1)^{\mathrm{wt}(\boldsymbol{c})} \mathcal{F}(\mathcal{D}_{\boldsymbol{a}} f + \boldsymbol{c}\cdot\boldsymbol{x}),\ \boldsymbol{c} \preceq \boldsymbol{a}. \tag{56}$$

There are 3^m coefficients, $U_{\boldsymbol{a},\boldsymbol{c}}$, where $\boldsymbol{c} \preceq \boldsymbol{a}$, but only 2^m complete autocorrelation profiles that we can obtain from $U_{\boldsymbol{a},\boldsymbol{c}}$ as each value is represented $2^{\mathrm{wt}(\overline{\boldsymbol{a}})}$ times to realise a complete set of 2^{2m} autocorrelation coefficients. Combining Eq. (20) with Eq. (55) and Eq. (56) yields

$$U_{\boldsymbol{a},\boldsymbol{e},\boldsymbol{r},\boldsymbol{\mu}} = 2^{-\mathrm{wt}(\boldsymbol{\theta})} \sum_{\boldsymbol{c} \in \boldsymbol{e}+V_{\boldsymbol{\theta}}} G_{\boldsymbol{a},\boldsymbol{c}} (-1)^{\overline{\boldsymbol{r}}\cdot\boldsymbol{c}},\ \boldsymbol{e} \preceq \boldsymbol{a} \preceq \boldsymbol{\mu},\ \boldsymbol{r} \preceq \boldsymbol{\theta}, \tag{57}$$

and

$$U_{\boldsymbol{a},\boldsymbol{c}} = (-1)^{\mathrm{wt}(\boldsymbol{c})} G_{\boldsymbol{a},\boldsymbol{c}},\ \boldsymbol{c} \preceq \boldsymbol{a}. \tag{58}$$

Note that the factor of $(-1)^{\mathrm{wt}(\boldsymbol{c})}$ is of no significance in this paper, but we retain it for completeness.

By combining Proposition 4 with Eq. (57) and Eq. (58) we can now express the fixed-aperiodic (nonmodular) autocorrelation coefficients in terms of the periodic/negaperiodic autocorrelation coefficients, and vice versa, where $\boldsymbol{e} \preceq \boldsymbol{a} \preceq \boldsymbol{\mu}$, $\boldsymbol{k} \preceq \boldsymbol{\mu}$, $\boldsymbol{\theta} = \boldsymbol{a} + \boldsymbol{\mu}$, and $\boldsymbol{r} = \boldsymbol{k} \,\&\, \boldsymbol{\theta}$

$$u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}} = 2^{-\mathrm{wt}(\boldsymbol{a})} \sum_{\boldsymbol{e} \preceq \boldsymbol{a}} U_{\boldsymbol{a},\boldsymbol{e},\boldsymbol{r},\boldsymbol{\mu}} (-1)^{\overline{\boldsymbol{k}}\cdot\boldsymbol{e}},\ \boldsymbol{k} \preceq \boldsymbol{\mu} \tag{59}$$

$$U_{\boldsymbol{a},\boldsymbol{e},\boldsymbol{r},\boldsymbol{\mu}} = \sum_{\boldsymbol{k} \preceq \boldsymbol{r}+V_{\boldsymbol{a}}} u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}} (-1)^{\boldsymbol{e}\cdot\overline{\boldsymbol{k}}},\ \boldsymbol{e} \preceq \boldsymbol{a} \tag{60}$$

$$u_{\boldsymbol{a},\boldsymbol{k}} = 2^{-\mathrm{wt}(\boldsymbol{a})} \sum_{\boldsymbol{c} \preceq \boldsymbol{a}} U_{\boldsymbol{a},\boldsymbol{c}} (-1)^{\overline{\boldsymbol{k}}\cdot\boldsymbol{c}},\ \boldsymbol{k} \preceq \boldsymbol{a} \tag{61}$$

$$U_{\boldsymbol{a},\boldsymbol{c}} = \sum_{\boldsymbol{k} \preceq \boldsymbol{a}} u_{\boldsymbol{a},\boldsymbol{k}} (-1)^{\boldsymbol{c}\cdot\overline{\boldsymbol{k}}},\ \boldsymbol{c} \preceq \boldsymbol{a}. \tag{62}$$

We now explain why Eq. (55) and Eq. (56) can be viewed as periodic/negaperiodic (modular) metrics.

Proposition 20. *Each periodic/negaperiodic autocorrelation of Eq.* (55) *and Eq.* (56) *is specified after fixing a subspace (resp. without fixing) by the parameters* $\boldsymbol{a}, \boldsymbol{e}, \boldsymbol{r}, \boldsymbol{\mu}$ *(resp.* $\boldsymbol{a}, \boldsymbol{c}$*). For each setting of the parameters, the coefficients can be calculated using multivariate polynomial multiplications which are periodically modular for the variables identified by the "1" positions of* $\boldsymbol{a} \,\&\, \overline{\boldsymbol{e}}$ *(resp.* $\boldsymbol{a} \,\&\, \overline{\boldsymbol{c}}$*), and negaperiodically modular for the variables identified by the "1" positions of* $\boldsymbol{e}$ *(resp.* $\boldsymbol{c}$*).*

Proof. Let $U_{\boldsymbol{a},\boldsymbol{c}}$ be as defined in Eq. (56), and let $\boldsymbol{z} \in \mathbb{C}^m$. Define $v(\boldsymbol{z})$, and $Q_{\boldsymbol{c}}(\boldsymbol{z})$ as follows

$$v(\boldsymbol{z}) = \sum_{\boldsymbol{x} \in \mathrm{GF}(2)^m} (-1)^{f(\boldsymbol{x})} \prod_{i \in \mathbb{Z}_m} z_i^{x_i} \tag{63}$$

$$Q_{\boldsymbol{c}}(\boldsymbol{z}) = \sum_{\boldsymbol{a} \in \mathrm{GF}(2)^m} U_{\boldsymbol{a},\boldsymbol{c}} \prod_{i \in \mathbb{Z}_m} z_i^{a_i}. \tag{64}$$

Then an expansion verifies the following modular relationship for $Q_{\boldsymbol{c}}(\boldsymbol{z})$

$$Q_{\boldsymbol{c}}(\boldsymbol{z}) = v(\boldsymbol{z}) v(\boldsymbol{z}^{-1}) \pmod{\prod_{i \in \mathbb{Z}_m} (z_i^2 - (-1)^{c_i})}. \tag{65}$$

$Q_{\boldsymbol{c}}(\boldsymbol{z})$ is the evaluation of a periodic (negaperiodic) multiplication for variable i if $c_i = 0$, (resp. $c_i = 1$). The above argument then carries over to Eq. (55) by first fixing the subspace $V_{\boldsymbol{\theta}}$, then computing all possible periodic/negaperiodic multivariate polynomial multiplications over the remaining unfixed subspace. □

We can recover the (nonmodular) polynomial $A(\boldsymbol{z})$ of Proposition 3 by applying the *Chinese remainder theorem* (CRT) to the residue polynomials $Q_{\boldsymbol{c}}(\boldsymbol{z})$. In summary,

$$\begin{aligned} A(\boldsymbol{z}) &= v(\boldsymbol{z}) v(\boldsymbol{z}^{-1}) = v(\boldsymbol{z}) v(\boldsymbol{z}^{-1}) \pmod{\prod_{i \in \mathbb{Z}_m} (z_i^4 - 1)} \\ &= \mathrm{CRT}(\{Q_{\boldsymbol{c}}(\boldsymbol{z})\}). \end{aligned} \tag{66}$$

In this way, we obtain an alternative derivation of Eq. (62). A similar argument can be used with respect to a fixed subspace, $V_{\boldsymbol{\theta}}$, so as to rederive Eq. (60).

B.2. Relationships to the Second Derivative

As $G_{a,c}$ is the Fourier spectrum of the first derivative of f, there is a natural relationship between the Fourier power spectra of $G_{a,c}$ and the *second derivative* of f, $\mathcal{D}_b\mathcal{D}_a f$, where $a, c, b \in \mathrm{GF}(2)^m$.

$$\sum_{c \preceq \mu} |G_{a,c}|^2 (-1)^{c \cdot k} = 2^{\mathrm{wt}(\mu)} \sum_{b \in k + V_{\overline{\mu}}} \mathcal{F}(\mathcal{D}_b \mathcal{D}_a f),\ k \preceq \mu. \tag{67}$$

Moreover we can use Parseval's theorem to establish the following.

$$\sum_{c \preceq \mu} |G_{a,c}|^4 = 2^{\mathrm{wt}(\mu)} \sum_{k \preceq \mu} \left(\sum_{b \in k + V_{\overline{\mu}}} \mathcal{F}(\mathcal{D}_b \mathcal{D}_a f) \right)^2. \tag{68}$$

Combining the above relationship with Eq. (23), we can establish the following upper bound on the *fixed-aperiodic sum-of-squares* with respect to a after fixing a subspace V_θ, referred to as $\sigma_{a,\mu}$, and defined in Eq. (24), in terms of the second derivative of f.

$$\sigma_{a,\mu} \leq 2^{-2\,\mathrm{wt}(\mu)} \sum_{k \preceq \mu} \left(\sum_{b \in k + V_{\overline{\mu}}} \mathcal{F}(\mathcal{D}_b \mathcal{D}_a f) \right)^2. \tag{69}$$

B.3. A Generalised Definition of APC

Using the results of this Appendix and Appendix C we are able to generalise Eq. (32) as follows.

$$\begin{aligned} u_{a,k,\mu} = 0,\ \forall k \preceq \mu \quad &\Leftrightarrow \quad U_{a,e,r,\mu} = 0,\ \forall e \preceq a,\ \forall r \preceq \theta \\ &\Leftrightarrow \quad G_{a,c} = 0,\ \forall c \preceq \mu \\ &\Leftrightarrow \quad \sum_{b \in k + V_{\overline{\mu}}} \mathcal{F}(\mathcal{D}_b \mathcal{D}_a f) = 0,\ \forall k \preceq \mu, \end{aligned} \tag{70}$$

where $a \preceq \mu$.

C. Symmetries of Aperiodic Autocorrelation

We summarise some important conditions for simplification of the fixed-aperiodic autocorrelation profile and and/or symmetry operations that operate on a Boolean function and that keep the multiset of fixed-aperiodic autocorrelation coefficients unchanged to within a multiplicative phase offset and to within a permutation of the coefficient positions within the autocorrelation profile.

C.1. Quadratic Simplification

When the degree of f is two, a substantial simplification of the fixed-aperiodic autocorrelation profile can be obtained as follows.

Lemma 21. *Let $f \in \mathcal{B}_m$ be a quadratic function, and let $u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}$ be as defined in Eq.* (13)*. Then, for any $\boldsymbol{k}' \preceq \boldsymbol{\mu}$, $u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}} = \pm u_{\boldsymbol{a},\boldsymbol{k}',\boldsymbol{\mu}}$.*

Proof. The proof is straightforward. □

The simplification described by this lemma significantly reduces the APC analysis for quadratic Boolean functions as we can set $k = 0$. From Section 4 the APC distance is equivalent to the distance measure for zero-dimensional QECCs. Such QECCs map to quadratic Boolean functions. As QECCs of the stabilizer type are conveniently described by self-dual additive codes over GF(4), quadratic Boolean functions with favourable APC can conversely be constructed with relative ease from self-dual additive codes over GF(4). This simplification implicitly exploits the symmetry of Lemma 21.

C.2. Index Permutation Symmetry (Hypergraph Isomorphism)

Lemma 22. *Define $f \in \mathcal{B}_m$. Let π be a permutation from $\mathbb{Z}_m$ to $\mathbb{Z}_m$. Let γ be a permutation from $\mathrm{GF}(2)^m$ to $\mathrm{GF}(2)^m$ such that, for $\boldsymbol{r} \in \mathrm{GF}(2)^m$, $\gamma(\boldsymbol{r})$ takes r_i to $r_{\pi(i)}$. For $f = f(x_0, x_1, \ldots, x_{m-1})$, let $f' = f(x_{\pi(0)}, x_{\pi(1)}, \ldots, x_{\pi(m-1)})$. Then $u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}(f') = u_{\gamma(\boldsymbol{a}),\gamma(\boldsymbol{k}),\gamma(\boldsymbol{\mu})}(f)$, so that both f and f' satisfy* APC(l) *of order q.*

C.3. Periodic and Negaperiodic Symmetries

The fixed-aperiodic autocorrelation coefficient magnitudes of a function $f \in \mathcal{B}_m$ remain unchanged to within a linear permutation of the indices after periodic and/or negaperiodic shift of the input variables of f. With $\boldsymbol{\gamma} \in \mathrm{GF}(2)^m$ define f' as a periodic shift of f, where $f'(\boldsymbol{x}) = f(\boldsymbol{x} + \boldsymbol{\gamma})$.

Proposition 23. *With $\boldsymbol{a}, \boldsymbol{k}, \boldsymbol{\gamma}, \boldsymbol{\mu} \in \mathrm{GF}(2)^m$, f' as defined above, and fixed-aperiodic autocorrelation coefficients as defined in Eq.* (13)*, $u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}(f) = u_{\boldsymbol{a},(\boldsymbol{k}+\boldsymbol{\gamma})\,\&\,\boldsymbol{\mu},\boldsymbol{\mu}}(f')$, where $\boldsymbol{k} \preceq \boldsymbol{\mu}$.*

Proof. Using Eq. (13), $u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}(f') = \mathcal{F}(\mathcal{D}_{\boldsymbol{a}} f' \phi_{\boldsymbol{k}+V_{\overline{\boldsymbol{\mu}}}}) = \mathcal{F}(\mathcal{D}_{\boldsymbol{a}} f \phi_{\boldsymbol{\gamma}+\boldsymbol{k}+V_{\overline{\boldsymbol{\mu}}}})$, where $\boldsymbol{k} \preceq \boldsymbol{\mu}$.

$$\begin{aligned}\boldsymbol{\gamma} + \boldsymbol{k} + V_{\overline{\boldsymbol{\mu}}} &= (\boldsymbol{\gamma} \,\&\, \boldsymbol{\mu} + \boldsymbol{k}) + \boldsymbol{\gamma} \,\&\, \overline{\boldsymbol{\mu}} + V_{\overline{\boldsymbol{\mu}}} \\ &= (\boldsymbol{\gamma} + \boldsymbol{k}) \,\&\, \boldsymbol{\mu} + (\boldsymbol{\gamma} \,\&\, \overline{\boldsymbol{\mu}} + V_{\overline{\boldsymbol{\mu}}}) \\ &= (\boldsymbol{\gamma} + \boldsymbol{k}) \,\&\, \boldsymbol{\mu} + V_{\overline{\boldsymbol{\mu}}},\ \boldsymbol{k} \preceq \boldsymbol{\mu}.\end{aligned}$$

After the change of variable $\boldsymbol{k}$ to $(\boldsymbol{k} + \boldsymbol{\gamma}) \,\&\, \boldsymbol{\mu}$, we obtain

$$u_{\boldsymbol{a},(\boldsymbol{k}+\boldsymbol{\gamma})\,\&\,\boldsymbol{\mu},\boldsymbol{\mu}}(f') = \mathcal{F}(\mathcal{D} f \phi_{\boldsymbol{k}+V_{\overline{\boldsymbol{\mu}}}}) = u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}(f),\ \boldsymbol{k} \preceq \boldsymbol{\mu}. \tag{71}$$

□

Similarly, with $\boldsymbol{\lambda} \in \mathrm{GF}(2)^m$ we define f'' as a negaperiodic shift of f, where $f''(\boldsymbol{x}) = f(\boldsymbol{x} + \boldsymbol{\lambda}) + \boldsymbol{\lambda} \cdot \boldsymbol{x} + \mathrm{wt}(\boldsymbol{\lambda})$.

Proposition 24. *With $\boldsymbol{a}, \boldsymbol{k}, \boldsymbol{\lambda}, \boldsymbol{\mu} \in \mathrm{GF}(2)^m$, f'' as defined above, and fixed-aperiodic autocorrelation coefficients as defined in Eq.* (13)

$$u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}(f) = (-1)^{\boldsymbol{\lambda}\cdot\boldsymbol{a}} u_{\boldsymbol{a},(\boldsymbol{k}+\boldsymbol{\lambda})\,\&\,\boldsymbol{\mu},\boldsymbol{\mu}}(f''), \tag{72}$$

where $\boldsymbol{k} \preceq \boldsymbol{\mu}$.

Proof. Remembering that f' is a periodic shift of f, observe that $\mathcal{D}_{\boldsymbol{a}} f'' = f(\boldsymbol{x} + \boldsymbol{\lambda}) + f(\boldsymbol{x} + \boldsymbol{\lambda} + \boldsymbol{a}) + \boldsymbol{\lambda} \cdot \boldsymbol{a} = \mathcal{D}_{\boldsymbol{a}} f' + \boldsymbol{\lambda} \cdot \boldsymbol{a}$. Therefore

$$\begin{aligned}u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}(f'') &= \mathcal{F}(\mathcal{D}_{\boldsymbol{a}} f'' \phi_{\boldsymbol{k}+V_{\overline{\boldsymbol{\mu}}}}) \\ &= \mathcal{F}(\mathcal{D}_{\boldsymbol{a}} f' \phi_{\boldsymbol{k}+V_{\overline{\boldsymbol{\mu}}}} + \boldsymbol{\lambda} \cdot \boldsymbol{a}) \\ &= (-1)^{\boldsymbol{\lambda}\cdot\boldsymbol{a}} \mathcal{F}(\mathcal{D}_{\boldsymbol{a}} f' \phi_{\boldsymbol{\lambda}+\boldsymbol{k}+V_{\overline{\boldsymbol{\mu}}}}),\end{aligned}$$

where $\boldsymbol{k} \preceq \boldsymbol{\mu}$. Substituting $\boldsymbol{k}$ with $(\boldsymbol{k} + \boldsymbol{\lambda}) \,\&\, \boldsymbol{\mu}$ gives $u_{\boldsymbol{a},(\boldsymbol{k}+\boldsymbol{\lambda})\,\&\,\boldsymbol{\mu},\boldsymbol{\mu}}(f'') = (-1)^{\boldsymbol{\lambda}\cdot\boldsymbol{a}} u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}(f)$, and the proposition follows. □

We can combine the above results for periodic/negaperiodic shift (Propositions 23 and 24) as follows. With $\boldsymbol{\gamma}, \boldsymbol{\lambda} \in \mathrm{GF}(2)^m$ we define f_{pn} as a periodic/negaperiodic shift of f.

$$f_{pn}(\boldsymbol{x}) = f(\boldsymbol{x} + \boldsymbol{\gamma}) + \boldsymbol{\lambda} \cdot \boldsymbol{x} + \mathrm{wt}(\boldsymbol{\lambda}), \tag{73}$$

where $\boldsymbol{\lambda} \preceq \boldsymbol{\gamma}$.

Proposition 25. *With $\boldsymbol{a}, \boldsymbol{k}, \boldsymbol{\gamma}, \boldsymbol{\lambda}, \boldsymbol{\mu} \in \mathrm{GF}(2)^m$, f_{pn} as defined above, and fixed-aperiodic autocorrelation coefficients as defined in Eq.* (13)

$$u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}(f) = (-1)^{\boldsymbol{\lambda}\cdot\boldsymbol{a}} u_{\boldsymbol{a},(\boldsymbol{k}+\boldsymbol{\gamma})\,\&\,\boldsymbol{\mu},\boldsymbol{\mu}}(f_{pn}), \tag{74}$$

where $\boldsymbol{k} \preceq \boldsymbol{\mu}$ and $\boldsymbol{\lambda} \preceq \boldsymbol{\gamma}$.

Proof. Combine Propositions 23 and 24. □

Corollary 26. *For the special case with $\boldsymbol{\gamma} \preceq \overline{\boldsymbol{\mu}}$ and f_{pn} defined as above, $u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}(f) = u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}(f_{pn})$, where $\boldsymbol{k} \preceq \boldsymbol{\mu}$.*

Proof. $\boldsymbol{\gamma}\,\&\,\boldsymbol{\mu} = 0$. □

It follows that a periodic shift (resp. negaperiodic shift) of f after fixing a subspace $V_{\boldsymbol{\theta}}$ does not change the values (resp. magnitudes) of the fixed-aperiodic autocorrelation coefficients of f, but may permute them.

Given f_{pn} as defined above, Eq. (13), and Proposition 4, we obtain the following identities for the periodic/negaperiodic autocorrelation coefficients given in Lemma 27.

Lemma 27.

$$G_{\boldsymbol{a},\boldsymbol{c}}(f) = (-1)^{\boldsymbol{\lambda}\cdot\boldsymbol{a}+\boldsymbol{\gamma}\cdot\boldsymbol{c}} G_{\boldsymbol{a},\boldsymbol{c}}(f_{pn}),\ \boldsymbol{\lambda} \preceq \boldsymbol{\gamma},\ \boldsymbol{c} \preceq \boldsymbol{\mu}, \tag{75}$$

$$U_{\boldsymbol{a},\boldsymbol{c}}(f) = (-1)^{\boldsymbol{\lambda}\cdot\boldsymbol{a}+\boldsymbol{\gamma}\cdot\boldsymbol{c}} U_{\boldsymbol{a},\boldsymbol{c}}(f_{pn}),\ \boldsymbol{\lambda} \preceq \boldsymbol{\gamma},\ \boldsymbol{c} \preceq \boldsymbol{a}, \tag{76}$$

$$U_{\boldsymbol{a},\boldsymbol{e},\boldsymbol{r},\boldsymbol{\mu}}(f) = (-1)^{\boldsymbol{\lambda}\cdot\boldsymbol{a}+\boldsymbol{\gamma}\cdot\boldsymbol{e}} U_{\boldsymbol{a},\boldsymbol{e},(\boldsymbol{r}+\boldsymbol{\gamma}\,\&\,\boldsymbol{\theta}),\boldsymbol{\mu}}(f_{pn}),\ \boldsymbol{\lambda} \preceq \boldsymbol{\gamma},\ \boldsymbol{e} \preceq \boldsymbol{a},\ \boldsymbol{r} \preceq \boldsymbol{\theta}. \tag{77}$$

Proof. For $\boldsymbol{k} \preceq \boldsymbol{\mu}$ and $\boldsymbol{\lambda} \preceq \boldsymbol{\gamma}$, and noting that, for $\boldsymbol{c} \preceq \boldsymbol{\mu}$, $\boldsymbol{\gamma}\,\&\,\boldsymbol{\mu}\cdot\boldsymbol{c} = \boldsymbol{\gamma}\cdot\boldsymbol{c}$,

$$\begin{aligned}(-1)^{\boldsymbol{\lambda}\cdot\boldsymbol{a}} u_{\boldsymbol{a},(\boldsymbol{k}+\boldsymbol{\gamma})\,\&\,\boldsymbol{\mu},\boldsymbol{\mu}} &= 2^{-\operatorname{wt}(\boldsymbol{\mu})}(-1)^{\boldsymbol{\lambda}\cdot\boldsymbol{a}} \sum_{\boldsymbol{c}\preceq\boldsymbol{\mu}} G_{\boldsymbol{a},\boldsymbol{c}}(-1)^{(\boldsymbol{k}+\boldsymbol{\gamma})\cdot\boldsymbol{c}} \\ &= 2^{-\operatorname{wt}(\boldsymbol{\mu})}(-1)^{\boldsymbol{\lambda}\cdot\boldsymbol{a}} \sum_{\boldsymbol{c}\preceq\boldsymbol{\mu}} ((-1)^{\boldsymbol{\gamma}\cdot\boldsymbol{c}} G_{\boldsymbol{a},\boldsymbol{c}})(-1)^{\boldsymbol{k}\cdot\boldsymbol{c}}.\end{aligned}$$

The results for $U_{\boldsymbol{a},\boldsymbol{c}}$ and $U_{\boldsymbol{a},\boldsymbol{e},\boldsymbol{r},\boldsymbol{\mu}}$ follow in a similar way. □

It follows that the magnitudes of the periodic/negaperiodic autocorrelation coefficients are unchanged by a periodic and/or negaperiodic shift of f to within a linear permutation of the indices.

As the magnitudes of $u_{\boldsymbol{a},\boldsymbol{k},\boldsymbol{\mu}}(f)$, $U_{\boldsymbol{a},\boldsymbol{c}}(f)$, and $U_{\boldsymbol{a},\boldsymbol{e},\boldsymbol{r},\boldsymbol{\mu}}$ are invariant to a periodic and/or negaperiodic shift of f to within a linear permutation,

it follows, from Eq. (26), Definition 6, and Eq. (32) that $\sigma_{\boldsymbol{a},\boldsymbol{\theta}}(f)$, $\mathcal{E}(f)$, $\sigma(f)$, and the APC of f are invariant to periodic and/or negaperiodic shifts of f. We summarise these observations in the following corollary.

Corollary 28. *For $f \in \mathcal{B}_m$, $\boldsymbol{\mu} \in \mathrm{GF}(2)^m$, and $\boldsymbol{a} \preceq \boldsymbol{\mu}$, let f_{pn} be a periodic and/or negaperiodic shift of f. Then $\sigma_{\boldsymbol{a},\boldsymbol{\mu}}(f_{pn}) = \sigma_{\boldsymbol{a},\boldsymbol{\mu}}(f)$, $\mathcal{E}(f_{pn}) = \mathcal{E}(f)$, and $\sigma(f_{pn}) = \sigma(f)$. The functions f and f_{pn} will also satisfy APC of order q of the same degree, and have the same APC distance.*

D. Generalised Differential Biases of State-of-the-Art S-Boxes

In this section we examine the worst-case (truncated) differential bias for a given input differential weight, with respect to periodic, aperiodic, and fixed-aperiodic autocorrelation, for selected state-of-the-art S-boxes. More precisely, we consider a function f (S-box) mapping $\mathrm{GF}(2)^m$ to $\mathrm{GF}(2)^n$, and comprising n m-variable functions, $f_i \in \mathcal{B}_m$, $0 \leq i < n$. Then we define the linear space of the S-box to be the set of functions, $\{g_{\boldsymbol{c}} \mid \boldsymbol{c} \in \mathrm{GF}(2)^n\}$, such that $g_{\boldsymbol{c}} = \boldsymbol{c} \cdot f$. We then compute, for a given S-box, the maximum bias over all functions in the set $\{g_{\boldsymbol{c}}\}$. The periodic bias at weight $|\boldsymbol{a}|$ is given by $\frac{2^m + |p_{\boldsymbol{a}}|}{2^{m+1}}$, the aperiodic bias at weight $|\boldsymbol{a}|$ is given by $\frac{2^{m-|\boldsymbol{a}|} + |u_{\boldsymbol{a},k}|}{2^{m-|\boldsymbol{a}|+1}}$, and the fixed-aperiodic bias at weight $\boldsymbol{\mu}$ is given by $\frac{2^{m-|\boldsymbol{\mu}|} + |u_{\boldsymbol{a},k,\boldsymbol{\mu}}|}{2^{m-|\boldsymbol{\mu}|+1}}$, where, for a given differential weight, it always holds that the periodic bias is less than the aperiodic bias, which again is less than the fixed-aperiodic bias. Tables 2 and 3 show the results. For example, an exhaustive search of all 256 8-variable Boolean functions constructed by linear combinations of the 8 constituent Boolean functions of the AES S-box reveals that a weight-4 differential can be found with bias 0.56, 0.94, and 1.00, for the periodic, aperiodic, and fixed-aperiodic differentials, respectively.

Table 2: *Periodic (P), Aperiodic (A), and Fixed-Aperiodic (F) Autocorrelation Biases for Selected S-Boxes*

		Differential Weight								
S-box		1	2	3	4	5	6	7	8	9
AES [37]	P	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	
(8×8)	A	0.56	0.66	0.81	0.94	1.00	1.00	1.00	1.00	
	F	0.56	0.66	0.81	1.00	1.00	1.00	1.00	1.00	
Khazad [38]	P	0.67	0.67	0.69	0.70	0.67	0.67	0.66	0.63	
(8×8)	A	0.67	0.77	0.94	1.00	1.00	1.00	1.00	1.00	
	F	0.67	0.77	0.94	1.00	1.00	1.00	1.00	1.00	
Whirlpool [39]	P	0.66	0.69	0.67	0.69	0.66	0.67	0.66	0.64	
(8×8)	A	0.66	0.75	0.84	1.00	1.00	1.00	1.00	1.00	
	F	0.66	0.78	0.91	1.00	1.00	1.00	1.00	1.00	
Misty1 [40]	P	0.56	0.56	0.56	0.56	0.56	0.56	0.56		
(7×7)	A	0.56	0.75	0.75	1.00	1.00	1.00	1.00		
	F	0.56	0.75	1.00	1.00	1.00	1.00	1.00		
Misty1	P	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
(9×9)	A	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	F	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
DES-1 [41]	P	0.88	0.81	0.81	0.81	0.75	0.69			
(6×4)	A	0.88	0.94	1.00	1.00	1.00	1.00			
	F	0.88	1.00	1.00	1.00	1.00	1.00			
DES-2	P	0.94	0.81	0.81	0.81	0.88	0.75			
(6×4)	A	0.94	0.94	1.00	1.00	1.00	1.00			
	F	0.94	1.00	1.00	1.00	1.00	1.00			
DES-3	P	0.88	0.75	0.81	0.81	0.75	0.69			
(6×4)	A	0.88	0.88	1.00	1.00	1.00	1.00			
	F	0.88	1.00	1.00	1.00	1.00	1.00			
DES-4	P	1.00	0.75	0.75	1.00	1.00	0.75			
(6×4)	A	1.00	1.00	1.00	1.00	1.00	1.00			
	F	1.00	1.00	1.00	1.00	1.00	1.00			
DES-5	P	0.81	0.81	0.81	0.81	0.75	0.63			
(6×4)	A	0.81	0.94	1.00	1.00	1.00	1.00			
	F	0.81	1.00	1.00	1.00	1.00	1.00			
DES-6	P	0.81	0.88	0.81	0.81	0.81	0.69			
(6×4)	A	0.81	0.94	1.00	1.00	1.00	1.00			
	F	0.81	1.00	1.00	1.00	1.00	1.00			
DES-7	P	0.88	0.88	0.81	0.81	0.81	0.69			
(6×4)	A	0.88	1.00	1.00	1.00	1.00	1.00			
	F	0.88	1.00	1.00	1.00	1.00	1.00			
DES-8	P	0.88	0.88	0.81	0.81	0.75	0.75			
(6×4)	A	0.88	0.94	1.00	1.00	1.00	1.00			
	F	0.88	1.00	1.00	1.00	1.00	1.00			

Table 3: *Periodic (P), Aperiodic (A), and Fixed-Aperiodic (F) Autocorrelation Biases for Selected S-Boxes*

		Differential Weight					
S-box		1	2	3	4	5	6
FDE-1 [42]	P	0.69	0.88	0.88	0.88	0.75	0.63
(6×4)	A	0.69	1.00	1.00	1.00	1.00	1.00
	F	0.69	1.00	1.00	1.00	1.00	1.00
FDE-2	P	0.69	0.69	0.75	0.75	0.75	0.63
(6×4)	A	0.69	0.81	1.00	1.00	1.00	1.00
	F	0.69	0.88	1.00	1.00	1.00	1.00
FDE-3	P	0.75	0.75	0.75	0.69	0.69	0.75
(6×4)	A	0.75	0.88	1.00	1.00	1.00	1.00
	F	0.75	0.88	1.00	1.00	1.00	1.00
FDE-4	P	0.81	0.75	0.81	0.81	0.75	0.63
(6×4)	A	0.81	0.88	1.00	1.00	1.00	1.00
	F	0.81	1.00	1.00	1.00	1.00	1.00
FDE-5	P	0.75	0.69	0.75	0.75	0.69	0.69
(6×4)	A	0.75	0.94	1.00	1.00	1.00	1.00
	F	0.75	0.94	1.00	1.00	1.00	1.00
FDE-6	P	0.75	0.75	0.75	0.75	0.75	0.63
(6×4)	A	0.75	0.81	1.00	1.00	1.00	1.00
	F	0.75	0.88	1.00	1.00	1.00	1.00
FDE-7	P	0.75	0.75	0.75	0.75	0.69	0.69
(6×4)	A	0.75	0.88	1.00	1.00	1.00	1.00
	F	0.75	0.88	1.00	1.00	1.00	1.00
FDE-8	P	0.69	0.75	0.75	0.81	0.75	0.63
(6×4)	A	0.69	0.88	1.00	1.00	1.00	1.00
	F	0.69	0.88	1.00	1.00	1.00	1.00
$[[6, 0, 4]]$ Hexacode	P	0.50	0.50	0.50	1.00	0.50	0.50
(single function)	A	0.50	0.50	0.50	1.00	0.50	1.00
	F	0.50	0.50	0.50	1.00	1.00	1.00

REFERENCES

[1] PRENEEL, B., VAN LEEKWIJCK, W., VAN LINDEN, L., GOVAERTS, R., VANDEWALLE, J.: Propagation characteristics of Boolean functions. In *Advances in Cryptology – EUROCRYPT '90*, *Lecture Notes in Comput. Sci.*, vol. 473, pp. 161–173. Springer, Berlin, 1991.

[2] CARLET, C.: On cryptographic propagation criteria for Boolean functions. *Inform. and Comput.* 151(1–2), 32–56, 1999.

[3] KUROSAWA, K., SATOH, T.: Design of SAC/PC(l) of order k Boolean functions and three other cryptographic criteria. In *Advances in Cryptology – EUROCRYPT '97*, *Lecture Notes in Comput. Sci.*, vol. 1233, pp. 434–449. Springer, Berlin, 1997.

[4] HEIN, M., EISERT, J., BRIEGEL, H. J.: Multi-party entanglement in graph states. *Phys. Rev. A* 69(6), 2004. arXiv:quant-ph/0307130.

[5] PARKER, M. G., RIJMEN, V.: The quantum entanglement of binary and bipolar sequences. In *Sequences and Their Applications – SETA 2001*, Discrete Math. Theor. Comput. Sci., pp. 296–309. Springer, London, 2002. arXiv:quant-ph/0107106.

[6] BOUCHET, A.: Graphic presentations of isotropic systems. *J. Combin. Theory Ser. B* 45(1), 58–76, 1988.

[7] GLYNN, D. G.: On self-dual quantum codes and graphs, 2002. Submitted to Electron. J. Combin.

[8] GLYNN, D. G., GULLIVER, T. A., MAKS, J. G., GUPTA, M. K.: The geometry of additive quantum codes, 2005. Submitted to Springer.

[9] VAN DEN NEST, M., DEHAENE, J., DE MOOR, B.: Graphical description of the action of local Clifford transformations on graph states. *Phys. Rev. A* 69(2), 2004. arXiv:quant-ph/0308151.

[10] GULLIVER, T. A., PARKER, M. G.: The multivariate merit factor of a Boolean function. In *Proc. IEEE Information Theory Workshop on Coding and Complexity – ITW 2005*, pp. 58–62. 2005.

[11] PARKER, M. G.: Univariate and multivariate merit factors. In *Sequences and Their Applications – SETA 2004*, *Lecture Notes in Comput. Sci.*, vol. 3486, pp. 72–100. Springer, Berlin, 2005.

[12] DAVIS, J. A., JEDWAB, J.: Peak-to-mean power control in OFDM, Golay complementary sequences and Reed-Muller codes. *IEEE Trans. Inform. Theory* 45(7), 2397–2417, 1999.

[13] Gottesman, D.: *Stabilizer Codes and Quantum Error Correction*. Ph.D. thesis, Caltech, May 1997. arXiv:quant-ph/9705052.

[14] Grassl, M., Klappenecker, A., Rötteler, M.: Graphs, quadratic forms, and quantum codes. In *Proc. IEEE Int. Symp. Inform. Theory*, p. 45. 2002. arXiv:quant-ph/0703112.

[15] Riera, C., Parker, M. G.: Generalised bent criteria for Boolean functions (I). *IEEE Trans. Inform. Theory* 52(9), 4142–4159, 2006. arXiv:cs.IT/0502049.

[16] Schlingemann, D., Werner, R. F.: Quantum error-correcting codes associated with graphs. *Phys. Rev. A* 65(1), 012 308, 2002. arXiv:quant-ph/0012111.

[17] Tonchev, V. D.: Error-correcting codes from graphs. *Discrete Math.* 257(2–3), 549–557, 2002.

[18] Briegel, H. J., Raussendorf, R.: Persistent entanglement in arrays of interacting particles. *Phys. Rev. Lett.* 86(5), 910–913, 2001. arXiv:quant-ph/quant-ph/0004051.

[19] Raussendorf, R., Browne, D. E., Briegel, H. J.: Measurement-based quantum computation with cluster states. *Phys. Rev. A* 68(2), 2003. arXiv:quant-ph/0301052.

[20] Calderbank, A. R., Rains, E. M., Shor, P. M., Sloane, N. J. A.: Quantum error correction via codes over GF(4). *IEEE Trans. Inform. Theory* 44(4), 1369–1387, 1998. arXiv:quant-ph/9608006.

[21] Barnum, H., Linden, N.: Monotones and invariants for multi-particle quantum states. *J. Phys. A* 34(35), 6787–6805, 2001. arXiv:quant-ph/quant-ph/0103155.

[22] Charpin, P., Pasalic, E.: On propagation characteristics of resilient functions. In *Selected Areas in Cryptography*, *Lecture Notes in Comput. Sci.*, vol. 2595, pp. 175–195. Springer, Berlin, 2003.

[23] Evertse, J.-H.: Linear structures in block ciphers. In *Advances in Cryptology – EUROCRYPT '87*, *Lecture Notes in Comput. Sci.*, vol. 304, pp. 249–266. Springer, Berlin, 1987.

[24] Dillon, J. F.: *Elementary Hadamard Difference Sets*. Ph.D. thesis, Univ. of Maryland, 1974.

[25] Danielsen, L. E.: *On Self-Dual Quantum Codes, Graphs, and Boolean Functions*. Master's thesis, Dept. Informat., Univ. Bergen, Norway, Mar. 2005. arXiv:quant-ph/0503236.

[26] Danielsen, L. E., Parker, M. G.: Spectral orbits and peak-to-average power ratio of Boolean functions with respect to the $\{I, H, N\}^n$ transform.

In *Sequences and Their Applications – SETA 2004*, *Lecture Notes in Comput. Sci.*, vol. 3486, pp. 373–388. Springer, Berlin, 2005. arXiv:cs.IT/0504102.

[27] Höhn, G.: Self-dual codes over the Kleinian four group. *Math. Ann.* 327(2), 227–255, 2003. arXiv:math.CO/0005266.

[28] Danielsen, L. E., Parker, M. G.: On the classification of all self-dual additive codes over GF(4) of length up to 12. *J. Combin. Theory Ser. A* 113(7), 1351–1367, 2006. arXiv:math.CO/0504522.

[29] Sloane, N. J. A.: The On-Line Encyclopedia of Integer Sequences. `http://www.research.att.com/~njas/sequences/`.

[30] McKay, B. D.: *nauty User's Guide, Version 2.2*, 2003. `http://cs.anu.edu.au/~bdm/nauty/`.

[31] Parker, M. G.: Generalised S-box nonlinearity. Public Document NES/DOC/UIB/WP5/020/A, NESSIE, 2003.

[32] Klappenecker, A., Rötteler, M.: Clifford codes. In *Mathematics of quantum computation*, Comput. Math. Ser., pp. 253–273. Chapman & Hall/CRC Press, Boca Raton, FL, 2002.

[33] Canteaut, A., Videau, M.: Degree of composition of highly nonlinear functions and applications to higher order differential cryptanalysis. In *Advances in Cryptology – EUROCRYPT 2002*, *Lecture Notes in Comput. Sci.*, vol. 2332, pp. 518–533. Springer, Berlin, 2002.

[34] Standaert, F.-X., Rouvroy, G., Piret, G., Quisquater, J.-J., Legat, J.-D.: Key-dependent approximations in cryptanalysis. In *Proc. 24th Symp. on Inform. Theory in the Benelux*. 2003.

[35] Chabaud, F., Vaudenay, S.: Links between differential and linear cryptanalysis. In *Advances in Cryptology – EUROCRYPT '94*, *Lecture Notes in Comput. Sci.*, vol. 950, pp. 356–365. Springer, Berlin, 1995.

[36] Canteaut, A., Charpin, P.: Decomposing bent functions. *IEEE Trans. Inform. Theory* 49(8), 2004–2019, 2003.

[37] Daemen, J., Rijmen, V.: The block cipher Rijndael. NIST AES homepage, Feb. 2003. `http://www.nist.gov/aes/`.

[38] Rijmen, V., Barreto, P. S. L. M.: The KHAZAD legacy-level block cipher. In *First open NESSIE Workshop, Leuven*. 2000.

[39] Barreto, P. S. L. M., Rijmen, V.: The WHIRLPOOL hashing function. In *First open NESSIE workshop, Leuven*. 2000.

[40] Matsui, M.: New block encryption algorithm MISTY. In *Fast Software Encryption – FSE '97*, *Lecture Notes in Comput. Sci.*, vol. 1267, pp. 54–68.

Springer, Berlin, 1997.

[41] Data encryption standard. FIPS Publication 46, National Bureau of Standards, U.S. Dept. of Commerce, 1977.

[42] Shafieinezhad, A., Hendessi, F., Gulliver, T. A.: A structure for fast data encryption. *Int. J. Contemp. Math. Sci.* 2(29), 1401–1424, 2007.

PAPER VI

INTERLACE POLYNOMIALS: CLASSIFICATION, UNIMODALITY, AND CONNECTIONS TO CODES

Lars Eirik Danielsen Matthew G. Parker

INTERLACE POLYNOMIALS: ENUMERATION, UNIMODALITY, AND CONNECTIONS TO CODES

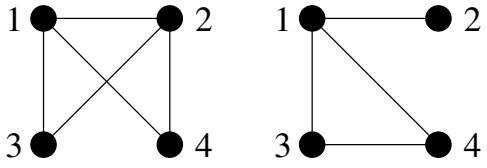
Lars Eirik Danielsen* Matthew G. Parker*

The interlace polynomial q was introduced by Arratia, Bollobás, and Sorkin. It encodes many properties of the orbit of a graph under edge local complementation (ELC). The interlace polynomial Q, introduced by Aigner and van der Holst, similarly contains information about the orbit of a graph under local complementation (LC). We have previously classified LC and ELC orbits, and now give an enumeration of the corresponding interlace polynomials of all graphs of order up to 12. An enumeration of all circle graphs of order up to 12 is also given. We show that there exist graphs of all orders greater than 9 with interlace polynomials q whose coefficient sequences are non-unimodal, thereby disproving a conjecture by Arratia et al. We have verified that for graphs of order up to 12, all polynomials Q have unimodal coefficients. It has been shown that LC and ELC orbits of graphs correspond to equivalence classes of certain error-correcting codes and quantum states. We show that the properties of these codes and quantum states are related to properties of the associated interlace polynomials.

1 INTRODUCTION

A *graph* is a pair $G = (V, E)$ where V is a set of *vertices*, and $E \subseteq V \times V$ is a set of *edges*. We will only consider *simple undirected* graphs, i.e.,

*Department of Informatics, University of Bergen, PB 7803, N-5020 Bergen, Norway.



(a) *The Graph G* **(b)** *The Graph* $G * 1$

Fig. 1: *Example of Local Complementation*

graphs where all edges are bidirectional and no vertex can be adjacent to itself. The *neighbourhood* of $v \in V$, denoted $N_v \subset V$, is the set of vertices connected to v by an edge. The number of vertices adjacent to v is called the *degree* of v. An *Eulerian graph* is a graph where all vertices have even degree. The *induced subgraph* of G on $W \subseteq V$ contains vertices W and all edges from E whose endpoints are both in W. The *complement* of G is found by replacing E with $V \times V - E$, i.e., the edges in E are changed to non-edges, and the non-edges to edges. Two graphs $G = (V, E)$ and $G' = (V, E')$ are *isomorphic* if and only if there exists a permutation π on V such that $\{u, v\} \in E$ if and only if $\{\pi(u), \pi(v)\} \in E'$. A *path* is a sequence of vertices, $(v_1, v_2, \ldots, v_i)$, such that $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{i-1}, v_i\} \in E$. A graph is *connected* if there is a path from any vertex to any other vertex in the graph. A graph is *bipartite* if its set of vertices can be decomposed into two disjoint sets such that no two vertices within the same set are adjacent. A *complete graph* is a graph where all pairs of vertices are connected by an edge. A *clique* is a complete subgraph. A k-clique is a clique consisting of k vertices. An *independent set* is the complement of a clique, i.e., an empty subgraph. The *independence number* of G is the size of the largest independent set in G.

Definition 1 ([1–3])**.** Given a graph $G = (V, E)$ and a vertex $v \in V$, let $N_v \subset V$ be the neighbourhood of v. *Local complementation* (LC) on v transforms G into $G * v$ by replacing the induced subgraph of G on N_v by its complement. (Fig. 1)

Definition 2 ([3])**.** Given a graph $G = (V, E)$ and an edge $\{u, v\} \in E$, *edge local complementation* (ELC) on $\{u, v\}$ transforms G into $G^{(uv)} = G * u * v * u = G * v * u * v$.

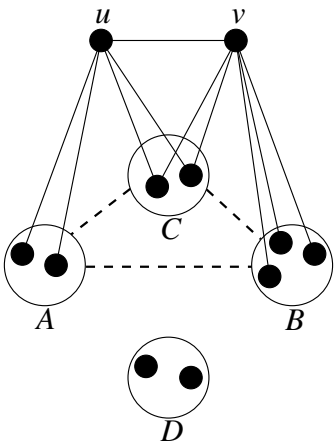


Fig. 2: *Visualization of the ELC Operation*

Definition 3 ([3])**.** ELC on $\{u, v\}$ can equivalently be defined as follows. Decompose $V \setminus \{u, v\}$ into the following four disjoint sets, as visualized in Fig. 2.

A Vertices adjacent to u, but not to v.

B Vertices adjacent to v, but not to u.

C Vertices adjacent to both u and v.

D Vertices adjacent to neither u nor v.

To obtain $G^{(uv)}$, perform the following procedure. For any pair of vertices $\{x, y\}$, where x belongs to class A, B, or C, and y belongs to a different class A, B, or C, "toggle" the pair $\{x, y\}$, i.e., if $\{x, y\} \in E$, delete the edge, and if $\{x, y\} \notin E$, add the edge $\{x, y\}$ to E. Finally, swap the labels of vertices u and v.

Definition 4. The *LC orbit* of a graph G is the set of all unlabeled graphs that can be obtained by performing any sequence of LC operations on G. Similarly, the *ELC orbit* of G comprises all unlabeled graphs that can be obtained by performing any sequence of ELC operations on G.

The LC operation was first defined by de Fraysseix [1], and later studied by Fon-der-Flaas [2] and Bouchet [3]. Bouchet defined ELC as

"complementation along an edge" [3], but this operation is also known as *pivoting* on a graph [4].

The recently defined *interlace polynomials* are based on the LC and ELC operations. Arratia, Bollobás, and Sorkin [4] defined the interlace polynomial $q(G)$ of the graph G. This work was motivated by a problem related to DNA sequencing [5].

Definition 5 ([4])**.** For every graph G, there is an associated interlace polynomial $q(G, x)$, which we will usually denote $q(G)$ for brevity. For the edgeless graph of order n, $E_n = (V, \emptyset)$, $q(E_n) = x^n$. For any other graph $G = (V, E)$, choose an arbitrary edge $\{u, v\} \in E$, and let

$$q(G) = q(G \setminus u) + q(G^{(uv)} \setminus u),$$

where $G \setminus u$ is the graph G with vertex u and all edges incident on u removed.

It was proven by Arratia et al. [4] that the polynomial is independent of the order of removal of edges, and that the polynomial is invariant under ELC, i.e., that $q(G) = q(G^{(uv)})$ for any edge $\{u, v\}$.

Aigner and van der Holst [6] later defined the interlace polynomial $Q(G)$ which similarly encodes properties of the LC orbit of G.

Definition 6 ([6])**.** For every graph G, there is an associated interlace polynomial $Q(G, x)$, which we will usually denote $Q(G)$ for brevity. For the edgeless graph of order n, $E_n = (V, \emptyset)$, $Q(E_n) = x^n$. For any other graph $G = (V, E)$, choose an arbitrary edge $\{u, v\} \in E$, and let

$$Q(G) = Q(G \setminus u) + Q(G^{(uv)} \setminus u) + Q(G * u \setminus u).$$

Again, the order of removal of edges is irrelevant, and the polynomial is invariant under LC and ELC. It was shown by Aigner and van der Holst [6] that both $q(G)$ and $Q(G)$ can also be derived from the ranks of matrices obtained by certain modifications of the adjacency matrix of G. A similar approach, but expressed in terms of certain sets of *local unitary transforms*, was shown by Riera and Parker [7]. If G is an unconnected graph with components G_1 and G_2, then $q(G) = q(G_1)q(G_2)$ and $Q(G) = Q(G_1)Q(G_2)$.

The interlace polynomials $q(G)$ and $Q(G)$ summarise several properties of the ELC and LC orbits of the graph G. The degree of the lowest-degree term of $q(G)$ equals the number of connected components of G, and is therefore one for a connected graph [4]. The degree

of $q(G)$ equals the maximum independence number of all graphs in the ELC orbit of G [6]. It follows that the degree of $q(G)$ is also an upper bound on the independence number of G. Likewise, the degree of $Q(G)$ gives the size of the largest independent set in the LC orbit of G [8]. The degree of $Q(G)$ will always be greater than or equal to the degree of $q(G)$. Evaluating interlace polynomials for certain values of x also gives us information about the associated graphs. For a graph G of order n, it always holds that $q(G,2) = 2^n$ and $Q(G,3) = 3^n$. $q(G,1)$ equals the number of induced subgraphs of G with an odd number of *perfect matchings* [6]. $Q(G,2)$ equals the number of general induced subgraphs of G (with possible loops attached to the vertices) with an odd number of general perfect matchings [6]. $Q(G,4)$ equals 2^n times the number of induced Eulerian subgraphs of G [6]. It has been shown that $q(G,-1) = (-1)^n 2^{n-r}$, where n is the order of G and r is the rank over GF(2) of $A+I$, where A is the adjacency matrix of G [6, 9]. $q(G,3)$ is always divisible by $q(G,1)$, and the quotient is an odd integer [6].

In their list of open problems [4], Arratia et al. pose the question of how many different interlace polynomials there are for graphs of order n. In Section 2, we answer this question for $n \leq 12$, for both interlace polynomials q and Q.

In the DNA sequencing setting [5], interlace polynomials of *circle graphs* are of particular interest. Arratia et al. [5] enumerated the circle graphs of order up to 9. In Section 3, we extend this enumeration to order 12.

Let $q(G) = a_1 x + a_2 x^2 + \cdots + a_d x^d$. Then the sequence of coefficients of q is $\{a_i\} = (a_1, a_2, \ldots, a_d)$. Arratia et al. [4] conjecture that this sequence is *unimodal* for all q. The sequence $\{a_i\}$ is unimodal if, for some $1 \leq k \leq d$, $a_i \leq a_j$ for all $i < j \leq k$, and $a_i \geq a_j$ for all $i > j \geq k$. In other words, the sequence is non-decreasing up to some coefficient k, and the rest of the sequence is non-increasing. In Section 4, we show that there exist interlace polynomials q whose coefficient sequences are non-unimodal, and thereby disprove the conjecture by Arratia et al. Our enumeration shows that all interlace polynomials of graphs of order up to 9 are unimodal, but that there are two graphs of order 10 with the same non-unimodal interlace polynomial. From these graphs of order 10 it is possible to construct graphs of any order greater than 10 with non-unimodal interlace polynomials. We verify that all interlace polynomials $Q(G)$ and all polynomials $xq(G, x+1)$ of graphs of order up to 12 have unimodal coefficients.

In Section 5 we highlight an interesting relationship between interlace polynomials, error-correcting codes, and quantum states. The LC orbit of a graph corresponds to the equivalence class of a *self-dual quantum code* [3, 10, 11], and the ELC orbit of a bipartite graph corresponds to the equivalence class of a *binary linear code* [12]. In both cases, the *minimum distance* of the code is equal to $\delta + 1$, where δ is the minimum vertex degree over all graphs in the corresponding orbit. We have previously shown [8] that a self-dual quantum code with high minimum distance often corresponds to a graph G where $\deg(Q)$, the degree of $Q(G)$, is small. A self-dual quantum code can also be interpreted as a *quantum graph state* [13]. A code with high minimum distance will correspond to a quantum state with a high degree of *entanglement*. The degree of $Q(G)$ gives an indicator of the entanglement in the graph state represented by G known as the *peak-to-average power ratio* [8] with respect to certain transforms. Another indicator of the entanglement in a graph state is the *Clifford merit factor* (CMF) [14], which can be derived from the evaluation of $Q(G)$ at $x = 4$ [15]. In Section 5 we give the range of possible values of δ, $\deg(Q)$, and $Q(G, 4)$ for graphs of order up to 12, and bounds on these parameters for graphs of order up to 25, derived from the best known self-dual quantum codes.

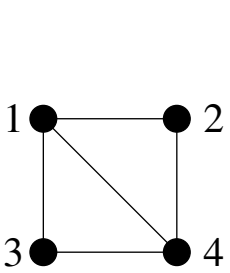
2 Enumeration of Interlace Polynomials

In the context of error-correcting codes, we have previously classified the LC orbits [16] and ELC orbits [12, 17] of all graphs on up to 12 vertices. In Table 1, the sequence $\{c_{L,n}\}$ gives the number of LC orbits of connected graphs on n vertices, while $\{t_{L,n}\}$ gives the total number of LC orbits of graphs on n vertices. Similarly, the sequence $\{c_{E,n}\}$ gives the number of ELC orbits of connected graphs on n vertices, while $\{t_{E,n}\}$ gives the total number of ELC orbits of graphs on n vertices. A database containing one representative from each LC orbit is available at `http://www.ii.uib.no/~larsed/vncorbits/`. A similar database of ELC orbits can be found at `http://www.ii.uib.no/~larsed/pivot/`.

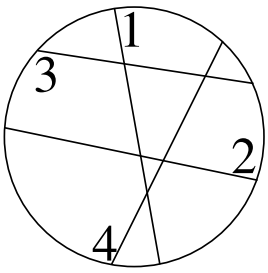
The question of how many distinct interlace polynomials there are for graphs of order n was posed by Arratia et al. [4]. For a representative from each LC and ELC orbit, we have calculated the interlace polynomials q and Q, respectively. We then counted the number of distinct interlace polynomials. In Table 2, the sequence $\{c_{Q,n}\}$ gives the number of interlace polynomials Q of connected graphs of order n, while $\{t_{Q,n}\}$ gives the total number of interlace polynomials Q of

Table 1: *Number of LC and ELC Orbits of Order n*

n	$c_{L,n}$	$t_{L,n}$	$c_{E,n}$	$t_{E,n}$
1	1	1	1	1
2	1	2	1	2
3	1	3	2	4
4	2	6	4	9
5	4	11	10	21
6	11	26	35	64
7	26	59	134	218
8	101	182	777	1068
9	440	675	6702	8038
10	3132	3990	104 825	114 188
11	40 457	45 144	3 370 317	3 493 965
12	1 274 068	1 323 363	231 557 290	235 176 097



(a) *The Circle Graph G*



(b) *The Circle Representation of G*

Fig. 3: *Example of a Circle Graph*

graphs of order n. Similarly, $\{c_{q,n}\}$ and $\{t_{q,n}\}$ give the numbers of interlace polynomials q. We observe that in Table 2, the relationship $t_{q,n} = c_{q,n} + t_{q,n-1}$ holds.

3 Enumeration of Circle Graphs

A graph G is a *circle graph* if each vertex in G can be represented as a chord on a circle, such that two chords intersect if and only if there is an edge between the two corresponding vertices in G. An example of a circle graph and its corresponding circle diagram is given in Fig. 3.

Table 2: *Number of Distinct Interlace Polynomials of Graphs of Order n*

n	$c_{Q,n}$	$t_{Q,n}$	$c_{q,n}$	$t_{q,n}$
1	1	1	1	1
2	1	2	1	2
3	1	3	2	4
4	2	6	4	8
5	4	11	9	17
6	10	24	24	41
7	23	52	71	112
8	84	152	257	369
9	337	521	1186	1555
10	2154	2793	7070	8625
11	22 956	26 178	56 698	65 323
12	486 488	515 131	614 952	680 275

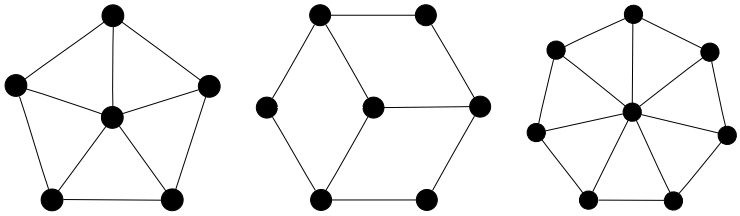


Fig. 4: *Circle Graph Obstructions*

Whether a given graph is a circle graph can be recognized in polynomial time [18]. It is also known that LC operations will map a circle graph to a circle graph, and a non-circle graph to a non-circle graph [19]. Bouchet [19] also proved that a graph G is a circle graph if and only if certain *obstructions*, shown in Fig. 4, do not appear as subgraphs anywhere in the LC orbit of G.

Arratia et al. [5] pointed out that an enumeration of circle graphs did not seem to have appeared in the literature before, and then gave an enumeration of circle graphs of order up to 9. Using our previous classification of LC orbits, and the fact that the circle graph property is preserved by LC operations, we are able to generate all circle graphs of order up to 12. In Table 3, the sequence $\{c_{c,n}\}$ gives the number of connected circle graphs of order n, while $\{t_{c,n}\}$ gives the total number of circle graphs of order n. The sequences $\{c'_{c,n}\}$ and $\{t'_{c,n}\}$ give the

Table 3: *Number of Circle Graphs on n Vertices*

n	$c_{c,n}$	$t_{c,n}$	$c'_{c,n}$	$t'_{c,n}$
1	1	1	1	1
2	1	2	1	2
3	2	4	1	3
4	6	11	2	6
5	21	34	4	11
6	110	154	10	25
7	789	978	23	55
8	8336	9497	81	157
9	117 283	127 954	293	499
10	2 026 331	2 165 291	1403	2059
11	40 302 425	42 609 994	7968	10 543
12	892 278 076	937 233 306	55 553	68 281

number of LC orbits containing circle graphs. A database with one representative from each LC orbit of connected circle graphs is available at `http://www.ii.uib.no/~larsed/circle/`.

4 Unimodality

Having calculated the interlace polynomials q of all graphs of order up to 12, it was possible to check whether their coefficient sequences were unimodal, as conjectured by Arratia et al. [4]. Note that a similar conjecture has been disproved for the related *Tutte polynomial* [20].

Our results show that all interlace polynomials q of graphs of order up to $n = 9$ are unimodal, but that for $n = 10$ there exists a single non-unimodal interlace polynomial with coefficient sequence $\{a_i\} = (2, 7, 6, 7, 4, 3, 2, 1, 0, 0)$. Only two graphs on 10 vertices, comprising a single ELC orbit, correspond to this polynomial. One of these graphs is shown in Fig. 5.

We have further found that, up to ELC equivalence, there are 4 graphs on 11 vertices with non-unimodal interlace polynomials, 3 of which are connected graphs, and 20 graphs on 12 vertices with non-unimodal polynomials, 15 of which are connected.

Given the single non-unimodal interlace polynomial of a graph of order $n = 10$, it is easy to show that there must exist non-unimodal interlace polynomials for all $n > 10$, since the following methods of

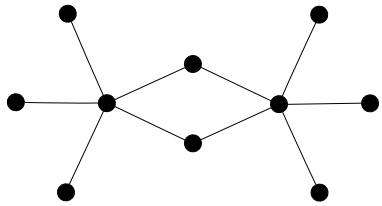


Fig. 5: *The Smallest Graph with Non-Unimodal Interlace Polynomial q*

extending a graph will preserve the non-unimodality of the associated interlace polynomial. Given a graph G on n vertices with non-unimodal interlace polynomial, we can add an isolated vertex to obtaining an unconnected graph G' on $n+1$ vertices, where $q(G') = xq(G)$ is clearly also non-unimodal. Non-unimodality is also preserved by *substituting* a vertex v of G by a clique of size m, i.e., we obtain G' where v is replaced by m vertices, all connected to each other and all connected to w whenever $\{v, w\}$ is an edge in G. It can then be shown that $q(G') = 2^m q(G)$ [4, Prop. 38].

Proposition 7. *Given a graph G, let G' be the graph obtained by* duplicating *a vertex v of G, i.e., by adding a vertex v' such that v' is connected to w whenever $\{v, w\}$ is an edge in G. The interlace polynomial of G can be written $q(G) = a(x) + cx^j + x^{j+1}b(x)$, where a and b are arbitrary polynomials, c is a constant, and $j = \deg(a) + 1$. The unimodality or lack thereof of G will be preserved in G' if $q(G \setminus v) = a(x) + x^j b(x)$.*

Proof. By duplicating the vertex v, we obtain a graph G' with interlace polynomial $q(G') = (1+x)q(G) - xq(G \setminus v)$ [4, Prop. 40]. If the condition above is satisfied, $q(G') = x^{j+2}a(x) + c(x^{j+1} + x^j) + b(x)$. The only difference between the coefficient sequences of $q(G)$ and $q(G')$ is that the coefficient c is repeated in $q(G')$, and unimodality or non-unimodality must therefore be preserved. □

Let G be the graph depicted in Fig. 5, and let v be one of the six vertices of degree one in this graph. If we duplicate v we obtain a graph whose interlace polynomial has the non-unimodal coefficient sequence $(2, 7, 6, 7, 6, 4, 3, 2, 1, 0, 0)$. According to Prop. 7, we can repeat the duplication of a vertex with degree one and the coefficient sequence will remain $(2, 7, 6, 7, 6, \ldots, 6, 4, 3, 2, 1, 0, 0)$, i.e., non-unimodal.

By the described extension methods we can obtain, from the single graph on 10 vertices shown in Fig. 5, all the 4 inequivalent graphs on

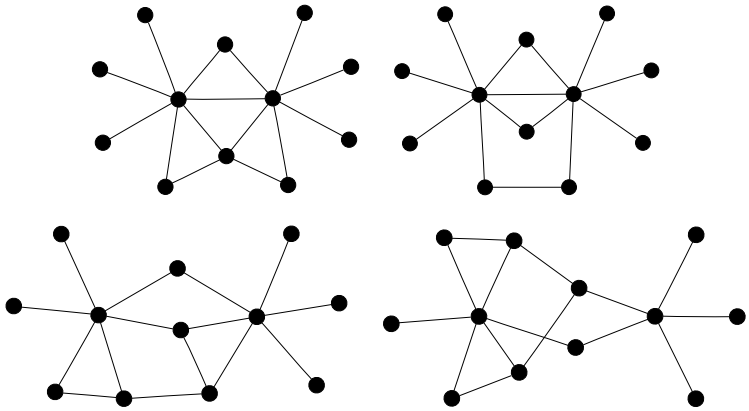


Fig. 6: *Non-trivial Graphs of Order 12 with Non-Unimodal Interlace Polynomial q*

11 vertices and 16 of the 20 inequivalent graphs on 12 vertices with non-unimodal interlace polynomials. Representatives from the ELC orbits of the 4 non-trivial graphs on 12 vertices with non-unimodal interlace polynomials are shown in Fig. 6.

The two following conjectures have been checked for all graphs on up to 12 vertices, and no counterexamples have been found.

Conjecture 8 ([4])**.** *For any interlace polynomial $q(G, x)$, the associated polynomial $xq(G, x + 1)$ has a unimodal coefficient sequence.*

Conjecture 9. *For any graph G, the interlace polynomial $Q(G)$ has a unimodal coefficient sequence.*

5 Connections to Codes and Quantum States

An important question is what the interlace polynomials $q(G)$ and $Q(G)$ actually compute about the graph G itself. When G is a circle graph, $q(G)$ can be used to solve counting problems relevant to DNA sequencing [5]. We will show that the interlace polynomials also give clues about the error-correction capability of codes and the entanglement of quantum states.

It is known that *self-dual quantum codes*, so called because they correspond to self-dual additive codes over $\mathrm{GF}(4)$ [21], can be represented as graphs [3, 10, 11, 22–24]. The LC orbit of a graph corresponds to the equivalence class of a self-dual quantum code [3, 10, 11]. Similarly, the

Table 4: *Range of* $\deg(Q)$ *For Given* δ *and* n

$\delta \backslash n$	2	3	4	5	6	7	8	9	10	11	12
1	1	2	2,3	3,4	3–5	3–6	3–7	4–8	4–9	4–10	4–11
2				2	3	3,4	3,4	3–5	4–6	4–7	4–8
3					2		3,4	3,4	3–5	4–6	4–7
4										4	4
5											4

ELC orbit of a bipartite graph corresponds to the equivalence class of a *binary linear code* [12]. In both cases the *minimum distance*, an important parameter that determines the error-correcting capability of the code, is equal to $\delta + 1$, where δ is the minimum vertex degree over all graphs in the corresponding LC or ELC orbit. A self-dual quantum code can also be interpreted as a *quantum graph state*, and the δ-value of the associated LC orbit is then an indicator of the degree of *entanglement* in the state.

Although the value δ can not be obtained from an interlace polynomial, several values that are correlated with δ are encoded in the interlace polynomial. The size of the largest independent set over all members of the LC orbit of G equals $\deg(Q)$, the degree of $Q(G)$ [6, 8]. We have previously shown that optimal self-dual quantum codes correspond to LC orbits where $\deg(Q)$ is small [8]. These codes have largest possible minimum distance for a given length n, and thus the associated LC orbits of graphs on n vertices have maximum possible values of δ. The data in Table 4 implies that the LC orbits with the highest δ-values also have the lowest values of $\deg(Q)$, but that the converse is not always true. In the context of quantum graph states, the value $2^{\deg(Q)}$ is equal to the *peak-to-average power ratio* [8] with respect to certain transforms, which is another indicator of the degree of entanglement in the state.

Another measure of the entanglement in a quantum graph state is the *Clifford merit factor* (CMF) [14]. A quantum graph state can be represented as a graph G, and the CMF of the state can be derived from the value obtained by evaluating the associated interlace polynomial $Q(G)$ at $x = 4$ [15]. The CMF value can be obtained with the formula $\frac{6^n}{2^n Q(G,4) - 6^n}$. Interestingly, $\frac{Q(G,4)}{2^n}$ also gives the number of induced Eulerian subgraphs of a graph on n vertices [6], which is invariant over all members of an LC orbit. As can be seen in Table 5, the LC orbits with the highest δ-values also have the lowest values of $Q(G, 4)$.

Table 5: *Range of* $\frac{Q(G,4)}{2^n}$ *For Given n and* δ

$n\backslash\delta$	1	2	3	4	5
2	3				
3	5				
4	8–9				
5	13–17	12			
6	20–33	19	18		
7	30–65	29–30			
8	47–129	45–48	44–45		
9	73–257	69–80	68–69		
10	112–513	106–128	104–109		
11	172–1025	160–183	157–180	156	
12	260–2049	244–362	237–288	238–239	234

Other evaluations of the interlace polynomials are also of interest in the context of quantum graph states, for instance $q(G,1)$ and $Q(G,2)$ give the number of *flat spectra* with respect to certain sets of transforms of the state [15].

Although no algorithm is known for computing the interlace polynomial of a graph efficiently, it is in general faster to generate interlace polynomials, by simply using the recursive algorithm given in Definitions 5 and 6, than it is to generate the entire ELC or LC orbits of a graph. We have calculated the interlace polynomials Q of graphs corresponding to the best known self-dual quantum codes, obtained from `http://www.codetables.de/` and from a search we have previously performed of *circulant graph codes* [25]. The results, for graphs of order n up to 25, are given in Table 6. Values printed in bold font are the best values we have found, and are thus lower bounds on the minimum possible values of $\deg(Q)$ and $Q(G,4)$ for the given n. The values of δ are known to be optimal, except for $n = 23$ and $n = 25$, where a graph with $\delta = 8$ could exist, and $n = 24$, $n = 26$, and $n = 27$, where $\delta = 9$ is possible. In general, the following bounds hold [21].

$$\delta \leq 2\left\lfloor \frac{n}{6} \right\rfloor + 1,$$

if the corresponding self-dual quantum code is of *Type II*, which means that its graph representation is *anti-Eulerian*, i.e., a graph where all vertices have odd degree. Such graphs must have an even number of

vertices, and it is interesting to note that the anti-Eulerian property is preserved by LC operations.

$$\delta \leq \begin{cases} 2\left\lfloor \frac{n}{6} \right\rfloor, & \text{if } n \equiv 0 \pmod 6 \\ 2\left\lfloor \frac{n}{6} \right\rfloor + 2, & \text{if } n \equiv 5 \pmod 6 \\ 2\left\lfloor \frac{n}{6} \right\rfloor + 1, & \text{otherwise,} \end{cases}$$

if the corresponding self-dual quantum code is of *Type I*, i.e., corresponds to a graph where at least one vertex has even degree.

For $n = 13$ and $n = 14$ we were able to compute the interlace polynomial Q of all graphs with optimal δ, since the corresponding codes have been classified [16, 26]. For other n, codes with the same δ but with lower $\deg(Q)$ or $Q(G,4)$ may exist. The best self-dual quantum codes correspond to LC orbits where δ is maximized, and our results for graphs on up to 12 vertices suggested that these LC orbits also minimize $\deg(Q)$ and $Q(G,4)$. However, in Table 6 we find several examples where the graph we have found with lowest $\deg(Q)$ does not have maximum δ. We have not found a single example where the lowest $Q(G,4)$ is found in a graph with suboptimal δ, which indicates that $Q(G,4)$ may be a better indicator of the distance of a code that $\deg(Q)$, and leads to the following conjecture.

Conjecture 10. *Let G be a graph on n vertices, and let δ be the minimum vertex degree over all graphs in the LC orbit of G. If there exists no other graph G' on n vertices such that $Q(G',4) < Q(G,4)$, then there exists no other graph on n vertices where the minimum vertex degree over all graphs in the LC orbit is greater than δ.*

Note that once we have found a graph G on n vertices with a certain $\deg(Q(G))$, we can obtain a graph G' on $n-1$ vertices with $\deg(Q(G')) = \deg(Q(G))$ or $\deg(Q(G')) = \deg(Q(G)) - 1$ by simply deleting any vertex of G. This process is equivalent to *shortening* a quantum code [27], and it is known that if the minimum vertex degree in the LC orbit of G is δ, then the minimum vertex degree in the LC orbit of G' is δ or $\delta - 1$.

A class of self-dual quantum codes known to have high minimum distance are the *quadratic residue codes*. The graphs corresponding to these codes are *Paley graphs*. To construct a Paley graph on n vertices, where n must be a prime power and $n \equiv 1 \pmod 4$, let the elements of the finite field $\mathrm{GF}(n)$ be the set of vertices, and let two vertices, i and j, be joined by an edge if and only if their difference is a quadratic residue

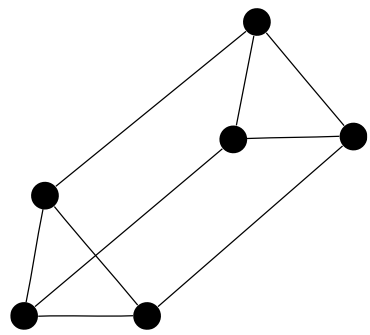
Table 6: *Best Found Values of* δ, $\deg(Q)$, *and* $\frac{Q(G,A)}{2^n}$ *From Quantum Codes*

n	δ	$\deg(Q)$	$\frac{Q(G,A)}{2^n}$
13	**4**	**4**	361
13	**4**	5	**360**
14	**5**	**4**	**549**
15	**5**	6	**830**
15	4	**5**	845
16	**5**	**5**	**1264**
17	**6**	6	**1872**
17	4	**5**	1890
18	**7**	6	**2808**
18	5	**5**	2835
19	**6**	**6**	**4296**
20	**7**	**6**	**6444**
21	**7**	9	**9672**
21	6	**6**	9756
22	**7**	**6**	**14 688**
23	**7**	7	**22 013**
23	6	**6**	22 235
24	**7**	**6**	**33 156**
25	**7**	**6**	**49 862**
25	**7**	7	**49 812**

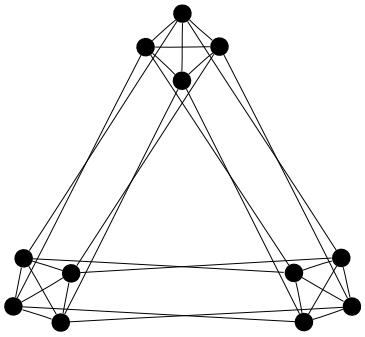
in GF(n)\{0}, i.e., there exists an $x \in$ GF(n)\{0} such that $x^2 \equiv i - j$. Paley graphs are known to have low independence numbers, and, since they correspond to strong quantum codes, the degrees of their interlace polynomials Q are also low, i.e., the size of the largest independent set in the LC orbit of a Paley graph is small, compared to other graphs on the same number of vertices. This suggests that Paley graphs, due to their high degree of symmetry, have the property that their independence numbers remain largely invariant with respect to LC. Another code construction is the *bordered quadratic residue code*, equivalent to extending a Paley graph by adding a vertex and connecting it to all existing vertices. For example, optimal quantum codes of length 5, 6, 29, and 30 can be constructed using Paley graphs or extended Paley graphs.

We have previously discovered [8] that many strong self-dual quantum codes can be represented as highly structured *nested clique graphs*. Some of these graphs are shown in Fig. 7. For instance, Fig. 7b shows a graph consisting of three 4-cliques. The remaining edges form a *Hamiltonian cycle*, i.e., a cycle that visits every vertex of the graph exactly once. Fig. 7c shows five 4-cliques interconnected by one Hamiltonian cycle and two cycles of length 10. Ignoring edges in the cliques, there are no cycles of length shorter than 5 in the graph. The graph in Fig. 7a can be viewed as two interconnected 3-cliques. Note that the graphs in Fig. 7 have values of δ, deg(Q), and $Q(G, 4)$ that match the optimal or best known values in Tables 4, 5, and 6. Also note that they are all δ-regular, which means that the number of edges is minimal for the given δ.

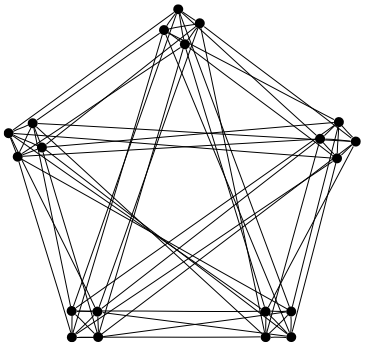
It is interesting to observe that the problem of finding good quantum codes, or highly entangled quantum states, can be reformulated as the problem of finding LC orbits of graphs with certain properties, and that these properties are related to the interlace polynomials of the graphs. Even though certain construction techniques are known, as shown above, many open problems remain, such as providing general bounds on δ, deg(Q), and $Q(G, 4)$, and finding new methods for constructing graphs with optimal or good values for these parameters. It would also be interesting to study possible connections between the observation that the best self-dual quantum codes have a minimal number of Eulerian subgraphs, and the fact that that many optimal self-dual quantum codes are of Type II, i.e., correspond to anti-Eulerian graphs. Note that all the graphs in Fig. 7 are anti-Eulerian. The graphs in Fig. 7 also give other clues as to the types of graphs that may optimise deg(Q) and



(a) $n = 6$, $\delta = 3$, $\deg(Q) = 2$, $\frac{Q(G,A)}{2^n} = 18$



(b) $n = 12$, $\delta = 5$, $\deg(Q) = 4$, $\frac{Q(G,A)}{2^n} = 234$



(c) $n = 20$, $\delta = 7$, $\deg(Q) = 6$, $\frac{Q(G,A)}{2^n} = 6444$

Fig. 7: *Examples of Nested Clique Graphs*

Table 7: *Number of LC Orbits Containing Connected Bipartite Graphs by δ and n*

$\delta \backslash n$	2	3	4	5	6	7	8	9	10	11	12
1	1	1	2	3	7	14	40	106	352	1218	5140
2					1	1	2	4	16	41	215
3							1		2	1	11
All	1	1	2	3	8	15	43	110	370	1260	5366

$Q(G,4)$. If a graph contains a k-clique, performing LC on any vertex in the clique will produce a graph with an independent set of size at least $k-1$. Thus the interlace polynomial Q of a complete graph will have the highest possible degree of any connected graph. This explains why our graphs contain several relatively small cliques. That the graphs contain a few long cycles reduces the number of cycles in the graph, which makes sense when we consider that a cycle is an Eulerian subgraph.

It is also possible to say something about which properties should not be present in a graph with optimal δ, $\deg(Q)$, or $Q(G,4)$. A bipartite graph on n vertices will have an independence number of at least $\lceil \frac{n}{2} \rceil$. Thus the interlace polynomial Q associated with an LC orbit that contains a bipartite graph will have degree at least $\lceil \frac{n}{2} \rceil$. Note that bipartiteness is preserved by ELC, but not by LC. In Table 7, we give the number of LC orbits containing connected bipartite graphs on n vertices with a given value of δ. Compare this to Table 9, which includes LC orbits of all connected graphs. It also turns out that circle graphs are bad. This is not surprising, given that the circle graph obstructions shown in Fig. 4 all have optimal values of δ. The obstruction on 6 vertices also has optimal value of $Q(4)$, and the two other obstructions have $Q(4)$ only one greater than optimal. In Table 8, we give the number of LC orbits of connected circle graphs on n vertices with a given value of δ.

ACKNOWLEDGEMENTS This research was supported by the Research Council of Norway.

Table 8: *Number of LC Orbits of Connected Circle Graphs by δ and n*

$\delta\backslash n$	2	3	4	5	6	7	8	9	10	11	12
1	1	1	2	3	9	21	75	277	1346	7712	54 067
2				1	1	2	5	16	55	254	1474
3							1		2	2	12
All	1	1	2	4	10	23	81	293	1403	7968	55 553

Table 9: *Number of LC Orbits of Connected Graphs by δ and n*

$\delta\backslash n$	2	3	4	5	6	7	8	9	10	11	12
1	1	1	2	3	9	22	85	363	2436	26 750	611 036
2				1	1	4	11	69	576	11 200	467 513
3					1		5	8	120	2506	195 455
4										1	63
5											1
All	1	1	2	4	11	26	101	440	3132	40 457	1 274 068

REFERENCES

[1] DE FRAYSSEIX, H.: Local complementation and interlacement graphs. *Discrete Math.* 33(1), 29–35, 1981.

[2] FON-DER FLAAS, D. G.: On local complementations of graphs. In *Combinatorics (Eger, 1987)*, *Colloq. Math. Soc. János Bolyai*, vol. 52, pp. 257–266. North-Holland, Amsterdam, 1988.

[3] BOUCHET, A.: Graphic presentations of isotropic systems. *J. Combin. Theory Ser. B* 45(1), 58–76, 1988.

[4] ARRATIA, R., BOLLOBÁS, B., SORKIN, G. B.: The interlace polynomial of a graph. *J. Combin. Theory Ser. B* 92(2), 199–233, 2004. arXiv:math.CO/0209045.

[5] ARRATIA, R., BOLLOBÁS, B., COPPERSMITH, D., SORKIN, G. B.: Euler circuits and DNA sequencing by hybridization. *Discrete Appl. Math.* 104(1–3), 63–96, 2000.

[6] AIGNER, M., VAN DER HOLST, H.: Interlace polynomials. *Linear Algebra Appl.* 377, 11–30, 2004.

[7] RIERA, C., PARKER, M. G.: Generalised bent criteria for Boolean functions (I). *IEEE Trans Inform. Theory* 52(9), 4142–4159, 2006. arXiv:cs.IT/0502049.

[8] Danielsen, L. E., Parker, M. G.: Spectral orbits and peak-to-average power ratio of Boolean functions with respect to the $\{I, H, N\}^n$ transform. In *Sequences and Their Applications – SETA 2004*, *Lecture Notes in Comput. Sci.*, vol. 3486, pp. 373–388. Springer, Berlin, 2005. arXiv:cs.IT/0504102.

[9] Balister, P. N., Bollobás, B., Cutler, J., Pebody, L.: The interlace polynomial of graphs at -1. *European J. Combin.* 23(7), 761–767, 2002.

[10] Van den Nest, M., Dehaene, J., De Moor, B.: Graphical description of the action of local Clifford transformations on graph states. *Phys. Rev. A* 69(2), 2004. arXiv:quant-ph/0308151.

[11] Glynn, D. G., Gulliver, T. A., Maks, J. G., Gupta, M. K.: The geometry of additive quantum codes, 2004. Submitted to Springer.

[12] Danielsen, L. E., Parker, M. G.: Edge local complementation and equivalence of binary linear codes, 2008. To appear in Des. Codes Cryptogr. arXiv:0710.2243.

[13] Hein, M., Eisert, J., Briegel, H. J.: Multi-party entanglement in graph states. *Phys. Rev. A* 69(6), 2004. arXiv:quant-ph/0307130.

[14] Parker, M. G.: Univariate and multivariate merit factors. In *Sequences and Their Applications – SETA 2004*, *Lecture Notes in Comput. Sci.*, vol. 3486, pp. 72–100. Springer, Berlin, 2005.

[15] Riera, C., Parker, M. G.: One and two-variable interlace polynomials: A spectral interpretation. In *Coding and Cryptography*, *Lecture Notes in Comput. Sci.*, vol. 3969, pp. 397–411. Springer, Berlin, 2006. arXiv:cs.IT/0504102.

[16] Danielsen, L. E., Parker, M. G.: On the classification of all self-dual additive codes over GF(4) of length up to 12. *J. Combin. Theory Ser. A* 113(7), 1351–1367, 2006. arXiv:math.CO/0504522.

[17] Riera, C., Parker, M. G.: On pivot orbits of Boolean functions. In *Fourth International Workshop on Optimal Codes and Related Topics*, pp. 248–253. Bulgarian Acad. Sci. Inst. Math. Inform., Sofia, 2005.

[18] Spinrad, J.: Recognition of circle graphs. *J. Algorithms* 16(2), 264–282, 1994.

[19] Bouchet, A.: Circle graph obstructions. *J. Combin. Theory Ser. B* 60(1), 107–144, 1994.

[20] Schwärzler, W.: The coefficients of the Tutte polynomial are not unimodal. *J. Combin. Theory Ser. B* 58(2), 240–242, 1993.

[21] Calderbank, A. R., Rains, E. M., Shor, P. M., Sloane, N. J. A.: Quantum error correction via codes over GF(4). *IEEE Trans. Inform. Theory* 44(4), 1369–1387, 1998. arXiv:quant-ph/9608006.

[22] Schlingemann, D., Werner, R. F.: Quantum error-correcting codes associated with graphs. *Phys. Rev. A* 65(1), 2002. arXiv:quant-ph/0012111.

[23] Schlingemann, D.: Stabilizer codes can be realized as graph codes. *Quantum Inf. Comput.* 2(4), 307–323, 2002. arXiv:quant-ph/0111080.

[24] Grassl, M., Klappenecker, A., Rötteler, M.: Graphs, quadratic forms, and quantum codes. In *Proc. IEEE Int. Symp. Inform. Theory*, p. 45. 2002. arXiv:quant-ph/0703112.

[25] Danielsen, L. E.: Graph-based classification of self-dual additive codes over finite fields, 2008. Preprint. arXiv:0801.3773.

[26] Varbanov, Z.: Some new results for additive self-dual codes over GF(4). *Serdica J. Comput.* 1(2), 213–227, 2007.

[27] Gaborit, P., Huffman, W. C., Kim, J.-L., Pless, V.: On additive GF(4) codes. In *Codes and Association Schemes*, *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, vol. 56, pp. 135–149. Amer. Math. Soc., Providence, RI, 2001.

PAPER VII

SELF-DUAL BENT FUNCTIONS

Claude Carlet Lars Eirik Danielsen
Matthew G. Parker Patrick Solé

SELF-DUAL BENT FUNCTIONS

Claude Carlet* Lars Eirik Danielsen†
Matthew G. Parker† Patrick Solé‡

A bent function is called self-dual if it is equal to its dual, and anti-self-dual if it is equal to the complement of its dual. We give a spectral characterization in terms of the Rayleigh quotient of the Sylvester Hadamard matrix, and derive an efficient search algorithm from this. Primary and secondary constructions are given. All self-dual bent Boolean functions of up to 6 variables and all quadratic such functions of 8 variables are classified, up to a restricted form of linear equivalence.

1 INTRODUCTION

Bent functions form a remarkable class of Boolean functions with applications in many domains, such as difference sets, spreading sequences for CDMA, error correcting codes, and cryptology. In symmetric cryptography, these functions can be used as building blocks of stream ciphers. They will not, in general, be used directly as combining functions or as filtering functions, because they are not balanced, but as Dobbertin showed [1], they can be used as an ingredient to build balanced filtering functions. While this class of Boolean functions is very small compared to the class of all Boolean functions, it is still large enough to make enumeration and classification impossible if the

*Department of Mathematics, University of Paris VIII, 2, rue de la Liberté; 93526 - Saint-Denis cedex 02, France.
†Department of Informatics, University of Bergen, PB 7803, N-5020 Bergen, Norway.
‡CNRS-I3S, Les Algorithmes, Euclide B, 2000 route des Lucioles, BP 121, 06 903 Sophia Antipolis, France.

number of variables is at least 10. It is therefore desirable to look for subclasses that are more amenable to generation, enumeration, and classification.

A subclass that has received little attention since Dillon's seminal thesis [2] is the subclass of those Boolean functions that are equal to their dual (or Fourier transform in Dillon's terminology). We call these *self-dual bent functions*. Of related interest are those bent functions whose dual is the complement of the function. We call these *anti-self-dual bent functions*. In this work we characterize the sign functions of these two class of functions as the directions where extrema of the *Rayleigh quotient* of the Sylvester type Hadamard matrix occur, or, equivalently, as eigenvectors of that matrix. This spectral characterization allows us to give a very simple and efficient search algorithm, which makes it possible to enumerate and classify all self-dual bent function of up to $n = 6$ variables and all quadratic such functions of $n = 8$ variables. The computational saving on the exhaustive search is doubly exponential in n. We derive primary constructions (Maiorana-McFarland and Dillon's partial spreads), secondary constructions (going from bent functions in n variables to self-dual or anti-self-dual bent functions in $n + m$ variables) and class symmetries (operations on Boolean functions that preserve self-duality or anti-self-duality). The subclass of the Maiorana-McFarland class of bent functions exhibits interesting connections with *self-dual codes*, a fact which was our original motivation at the start of the study: to connect the duality of codes with the duality of Boolean functions.

The material is organized as follows. Section 2 collects the notation and definitions that we need for the rest of the paper. Section 3 contains the characterization in terms of the Rayleigh quotient and the bounds on that quantity for an odd number of variables. Section 4 looks into constructions, first primary, then secondary. Section 5 describes the search algorithm and establishes the symmetry between self-dual and anti-self-dual bent functions. The numerical results are listed in Section 6.

2 Definitions and Notation

A *Boolean function* f of n variables is a map from $\mathrm{GF}(2)^n$ to $\mathrm{GF}(2)$. Its *sign function* is $F := (-1)^f$, and its *Walsh-Hadamard transform* (WHT)

can be defined as

$$\hat{F}(x) := \sum_{y \in \mathrm{GF}(2)^n} (-1)^{f(y)+x\cdot y}. \tag{1}$$

When F is viewed as a column vector, the matrix of the WHT is the Hadamard matrix H_n of Sylvester type, which we define by tensor products. Let

$$H := \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{2}$$

Let $H_n := H^{\otimes n}$ be the n-fold tensor product of H with itself, and let $\mathcal{H}_n := 2^{-\frac{n}{2}} H^{\otimes n}$ be its normalized version. Recall the Hadamard property,

$$H_n H_n^T = 2^n I_{2^n}, \tag{3}$$

where I_m is the $m \times m$ identity matrix. A Boolean function of n variables is said to be *bent* if and only if $\mathcal{H}_n F$ is the sign function of some other Boolean function. That function, denoted $\tilde{f}$, is called the *dual* of f. The sign function of $\tilde{f}$ is henceforth denoted $\tilde{F}$. If, furthermore, $f = \tilde{f}$, then f is *self-dual bent*. This means that its sign function is an eigenvector of $\mathcal{H}_n$ attached to the eigenvalue 1. Similarly, if $f = \tilde{f} + 1$ then f is *anti-self-dual bent*. This means that its sign function is an eigenvector of $\mathcal{H}_n$ attached to the eigenvalue -1.

3 Characterization

We define the *Rayleigh quotient* S_f of a Boolean function f of n variables by the character sum

$$S_f := \sum_{x,y \in \mathrm{GF}(2)^n} (-1)^{f(x)+f(y)+x\cdot y} = \sum_{x \in \mathrm{GF}(2)^n} F(x)\hat{F}(x) \tag{4}$$

Theorem 1. *Let n denote an even integer and f be a Boolean function of n variables. The modulus of the character sum S_f is at most $2^{\frac{3n}{2}}$ with equality if and only if f is self-dual bent or anti-self-dual bent.*

Proof. The triangle inequality yields

$$\left| \sum_{x,y} (-1)^{f(x)+f(y)+x\cdot y} \right| \leq \sum_{x} \left| \sum_{y} (-1)^{f(x)+f(y)+x.y} \right|$$

By the Cauchy-Schwarz inequality the latter sum is at most

$$\sqrt{2^n \sum_x \left(\sum_y (-1)^{f(x)+f(y)+x\cdot y} \right)^2},$$

which, by Parseval's identity $\left(\sum_x (\hat{F}(x))^2 = 2^{2n}\right)$, equals $2^{\frac{3n}{2}}$. So, $S_f \leq 2^{\frac{3n}{2}}$, with equality only if there is equality in these two inequalities. Equality holds in the Cauchy-Schwarz inequality if and only if $\left|F(x)\hat{F}(x)\right| = \left|\hat{F}(x)\right|$ is a constant function of x, i.e., if and only if f is bent. Equality in the triangle inequality then holds if and only if the sign of $F(x)\hat{F}(x) = 2^{\frac{n}{2}} F(x)\tilde{F}(x)$ is a constant function of x, i.e., if and only if f is also self-dual ($+$ sign) or anti-self-dual ($-$ sign). □

By using the sign function F of f we can write

$$S_f = \sum_{x \in \mathrm{GF}(2)^n} F(x)\hat{F}(x) = \langle F, H_n F \rangle . \tag{5}$$

The standard properties of the Rayleigh quotient attached to the real symmetric matrix H_n show that the maximum (resp. minimum) of S_f are obtained for F an eigenvector of H_n attached to a maximum (resp. minimum) eigenvalue of H_n, which are, by Proposition 2 below, $2^{\frac{n}{2}}$ (resp. $-2^{\frac{n}{2}}$). See for instance [3, p. 198] or any textbook in numerical analysis for basic definition and properties of the Rayleigh quotient of an Hermitian matrix. Alternatively, by using Proposition 2, the orthogonal decomposition in eigenspaces of H_n yields $F = F^+ + F^-$, with $F^\pm \in \mathrm{Ker}(H_n \pm 2^{\frac{n}{2}} I_{2^n})$, and $\langle F, F \rangle = \langle F^+, F^+ \rangle + \langle F^-, F^- \rangle$. Plugging this decomposition into S_f gives

$$S_f = 2^{\frac{n}{2}} \left\langle F^+, F^+ \right\rangle - 2^{\frac{n}{2}} \left\langle F^-, F^- \right\rangle , \tag{6}$$

and by the triangle inequality, $|S_f| \leq 2^{\frac{3n}{2}}$, with equality if and only if $F = F^+$ or $F = F^-$.

Proposition 2. *The Hamming distance between a self-dual bent function f_1 and an anti-self-dual bent function f_2, both of n variables, is 2^{n-1}.*

Proof. Let F_1 (resp. F_2) denote the sign function of f_1 (resp. f_2). On the one hand

$$\langle F_1, H_n F_2 \rangle = -2^{\frac{n}{2}} \langle F_1, F_2 \rangle ,$$

by anti-self-duality of f_2. On the other hand by self-adjunctness of H_n, we have

$$\langle F_1, H_n F_2 \rangle = \langle H_n F_1, F_2 \rangle ,$$

which equals $2^{\frac{n}{2}} \langle F_1, F_2 \rangle$, by self-duality of f_1. Since

$$\langle F_1, F_2 \rangle = - \langle F_1, F_2 \rangle = 0,$$

the result follows. □

An interesting open problem is to consider the maximum of S_f for n odd, when the eigenvectors of H_n cannot be in $\{\pm 1\}^n$.

Theorem 3. *The maximum Rayleigh quotient of a Boolean function g in an odd number of variables n is at least $S_g \geq 2^{\frac{3n-1}{2}}$.*

Proof. Let F be the sign function of a self-dual bent function in $n-1$ variables, so that $H_{n-1}F = 2^{\frac{n-1}{2}} F$. Define a Boolean function in n variables by its sign function $G = (F, F)$. Write $H_n = H \otimes H_{n-1}$, to derive

$$H_n G = (2H_{n-1}F, 0)^T = \left(2^{\frac{n+1}{2}} F, 0\right)^T .$$

Taking the dot product on the left by G yields

$$S_g = 2^{\frac{n+1}{2}} F^T F = 2^{\frac{n+1}{2}} 2^{n-1} = 2^{\frac{3n-1}{2}}.$$

□

4 Constructions

4.1 Primary Constructions

4.1.1 Maiorana-McFarland

A general class of bent functions is the *Maiorana-McFarland* class, i.e., functions of the form

$$x \cdot \phi(y) + g(y), \tag{7}$$

where x and y are variable vectors of dimension $\frac{n}{2}$, $\phi \in \mathrm{GL}(\frac{n}{2}, 2)$, and g is an arbitrary Boolean function. In the following theorem L^T denotes the transpose of L.

Theorem 4. *A Maiorana-McFarland function is self-dual bent (resp. anti-self-dual bent) if and only if $g(y) = b \cdot y + \epsilon$ and $\phi(y) = L(y) + a$, where L is a linear automorphism satisfying $L \times L^T = I_{\frac{n}{2}}$, $a = L(b)$, and a has even (resp. odd) Hamming weight. In both cases the code with parity check matrix $(I_{\frac{n}{2}}, L)$ is self-dual and (a, b) is one of its codewords. Conversely, such a Boolean function can be attached to the ordered pair (H, c), where H is the parity check matrix of a self-dual code of length n and c is one of its codewords.*

Proof. The dual of a Maiorana-McFarland bent function $x \cdot \phi(y) + g(y)$ is equal to $\phi^{-1}(x) \cdot y + g(\phi^{-1}(x))$ [4]. If the function f is self-dual then g and ϕ must be affine, that is, $g(y) = b \cdot y + \epsilon$ and $\phi(y) = L(y) + a$ (where L is a linear automorphism). Then f is self-dual if and only if, for every $x, y \in \mathrm{GF}(2)^{\frac{n}{2}}$, $x \cdot (L(y) + a) + b \cdot y + \epsilon = y \cdot L^{-1}(x + a) + L^{-1}(x + a) \cdot b + \epsilon$, that is, for every $x, y \in F_2^{\frac{n}{2}}$, $x \cdot L(y) = y \cdot L^{-1}(x)$, (i.e., $L \times L^T = I_n$), $a = L(b)$ and b has even weight. □

Any self-dual code of length n gives rise to K parity check matrices, and each such distinct parity check matrix gives rise to $2^{\frac{n}{2}-1}$ self-dual bent functions and $2^{\frac{n}{2}-1}$ anti-self-dual bent functions. Thus, any self-dual code of length n gives rise to $K \times 2^{\frac{n}{2}-1}$ self-dual bent functions and the same number of anti-self-dual bent functions, to within variable re-labelling. All such functions are quadratic. It is possible to classify and/or enumerate this class given a classification and/or enumeration of all self-dual codes, coupled with a method to classify and/or enumerate all distinct parity check matrices for each code. One way of performing this last task is to generate all *edge local complementation* (ELC) orbits [5], up to isomorphism, for the bipartite graphs associated with each distinct self-dual code of length n. For each of self-dual and anti-self-dual, enumeration would then be realised by summing the orbit sizes and then multiplying the result by $2^{\frac{n}{2}-1}$, and classification would be realised by listing each member in the union of orbits. Each member of such a list would then be a $\mathrm{RM}(2, n)$ coset leader for a coset of $2^{\frac{n}{2}-1}$ self-dual and $2^{\frac{n}{2}-1}$ anti-self-dual quadratic Boolean functions.

4.1.2 Dillon's Partial Spreads

Let $x, y \in \mathrm{GF}(2^{\frac{n}{2}})$. The class denoted $\mathcal{PS}_{ap}$ [4] consists of the so-called Dillon functions of the type

$$f(x, y) = g\left(\frac{x}{y}\right), \tag{8}$$

with the convention that $\frac{x}{y} = 0$ if $y = 0$, and where g is balanced and $g(0) = 0$.

Theorem 5. *A Dillon function is self-dual bent if g satisfies $g(1) = 0$, and, for all $u \neq 0$ the relation $g(u) = g(\frac{1}{u})$. There are exactly $\binom{2^{\frac{n}{2}-1}-1}{2^{\frac{n}{2}-2}}$ such functions.*

Proof. By [4] the dual of a Dillon function is obtained by exchanging the roles of x and y. Define g by its values on pairs $\{u, \frac{1}{u}\}$, for u different from zero and one. Counting and balancedness implies then that $g(1) = 0$ and that the number of such pairs where g takes the value one is $\binom{2^{\frac{n}{2}-1}-1}{2^{\frac{n}{2}-2}}$. The result follows. □

By complementing functions one may go beyond the $\mathcal{PS}_{ap}$ class.

Corollary 6. *Let g be a function from $\mathrm{GF}(2^{\frac{n}{2}})$ down to $\mathrm{GF}(2)$ that satisfies $g(1) = g(0)$, and, for all $u \neq 0$ the relation $g(u) = g(\frac{1}{u})$. If g is balanced with the same convention as above, the function $f(x, y) = g(\frac{x}{y})$ is self-dual bent.*

4.2 Secondary Constructions

4.2.1 Class Symmetries

In this section we give class symmetries, i.e., operations on Boolean functions that leave the self-dual bent class invariant as a whole. We define, following [6], the orthogonal group of index n over $\mathrm{GF}(2)$ as

$$\mathcal{O}_n := \{L \in \mathrm{GL}(n, 2) \mid LL^T = I_n\}. \tag{9}$$

Observe that $L \in \mathcal{O}_n$ if and only if (I_n, L) is the generator matrix of a self-dual binary code of length $2n$. Thus, for even n, an example is $I_n + J_n$, where J_n is the all-one matrix.

Theorem 7. *Let f denote a self-dual bent function of n variables. If $L \in \mathcal{O}_n$ and $c \in \{0, 1\}$ then $f(Lx) + c$ is self-dual bent.*

Proof. Set $g(x) := f(Lx) + c$. The Walsh-Hadamard transform of that function is

$$\hat{G}(x) = (-1)^c \hat{F}(L(x)) = (-1)^{f(Lx)+c} = (-1)^{g(x)},$$

where the first equality holds by a change of variable involving $L^{-1} = L^T$, and the next to the last by self-duality of f. □

Recall that a function is I-bent if it has flat spectrum with respect to some unitary transform U obtained by tensoring m matrices I_2 and $n-m$ matrices $\mathcal{H}_1$ in any order [7], for some $m \leq n$.

Theorem 8. *Let f denote a self-dual bent function in n variables that is also I-bent. Its I-bent dual is self-dual bent.*

Proof. By definition, there is a unitary matrix U and a Boolean function g such that $U(-1)^f = (-1)^g$. The result then follows from the fact that U commutes with $\mathcal{H}_n$.

$$\mathcal{H}_n(-1)^g = \mathcal{H}_n U(-1)^f = U\mathcal{H}_n(-1)^f = U(-1)^f,$$

where the last equality comes from the self-duality of f. □

4.2.2 $n+m$ Variables from n Variables and m Variables

For this subsection define the *duality* of a bent function to be 0 if it is self-dual bent and 1 if it is anti-self-dual bent. If f and g are Boolean functions in n and m variables, respectively, we define the *direct sum* of f and g as the Boolean function on $n+m$ variables given by $f(x)+g(y)$. The following result is immediate, and its proof is omitted. However, it shows that self-dual and anti-self-dual bent functions cannot be considered separately.

Proposition 9. *If f and g are bent functions of dualities ϵ and ν their direct sum is bent of duality $\epsilon+\nu$.*

A more general construction involving four functions can be found in [8]. If f_1, f_2 and g_1, g_2 are pairs of Boolean functions in n and m variables, respectively, define the *indirect sum* of these four functions by

$$h(x,y) := f_1(x) + g_1(y) + (f_1 + f_2(x))\,(g_1 + g_2(y))\,. \tag{10}$$

Theorem 10. *If f_1, f_2 (resp. g_1, g_2) are pairs of bent functions of dualities both ϵ (resp. both ν), their indirect sum is bent of duality $\epsilon+\nu$. If f_1 is bent, $f_2 = f_1^{\perp} + \epsilon$ for some $\epsilon \in \{0,1\}$, g_1 is self-dual bent, and g_2 is anti-self-dual bent, then the indirect sum of the four functions is self-dual bent of duality ϵ.*

Proof. The proof of the first assertion comes from the fact that the indirect sum is bent if all four functions are bent, and in this case the dual function is obtained as the indirect sum of the duals of the four functions [8]. Writing $f_i = f_i + \epsilon$, and $g_i = g_i + \nu$ for $i = 1, 2$, the result follows. The proof of the second assertion is similar and is omitted. □

As an example of the construction, take $g_1(y_1, y_2) = y_1 y_2$, which is self-dual bent, and $g_2(y_1, y_2) = y_1 y_2 + y_1 + y_2$, which is anti-self-dual bent. Let f be a bent function in n variables and let F (resp. $\tilde{F}$) be its sign function (resp. the sign function of its dual). The vector $(F, \tilde{F}, \tilde{F}, -F)$ is the sign function of a self-dual bent function in $n+2$ variables. The vector $(F, -\tilde{F}, -\tilde{F}, -F)$ is the sign function of an anti-self-dual bent function in $n+2$ variables. The observant reader will notice that the sign pattern of the above construction is the same as that of self-dual bent and anti-self-dual bent functions in 2 variables. This leads us to conjecture the existence of 20 different constructions of self-dual bent functions in $n+4$ variables from bent functions in n variables.

5 Search Algorithm

Theorem 11. *Let $n \geq 2$ be an even integer and Z be arbitrary in $\{\pm 1\}^{n-1}$. Define $Y := Z + \frac{2H_{n-1}}{2^{\frac{n}{2}}} Z$. If Y is in $\{\pm 1\}^{n-1}$, then the vector (Y, Z) is the sign function of a self-dual bent function in n variables.*

We prepare for the proof by a linear algebra lemma.

Lemma 12. *The spectrum of $\mathcal{H}_n$ consists of the two eigenvalues ± 1 with the same multiplicity 2^{n-1}. A basis of the eigenspace attached to 1 is formed of the rows of the matrix $(H_{n-1} + 2^{\frac{n}{2}} I_{2^{n-1}}, H_{n-1})$. An orthogonal decomposition of $\mathbb{R}^{2^n}$ in eigenspaces of H_n is*

$$\mathbb{R}^{2^n} = Ker(H_n + 2^{\frac{n}{2}} I_{2^n}) \oplus Ker(H_n - 2^{\frac{n}{2}} I_{2^n}).$$

Proof. (of Lemma 12) The minimal polynomial of $\mathcal{H}_n$ is $X^2 - 1$, by symmetry of $\mathcal{H}_n$ and the Hadamard property of H_n. Hence the spectrum. The multiplicity follows by $\mathrm{Tr}(\mathcal{H}_n) = 0$. The matrix $\mathcal{H}_n + I_n$ is a projector on the eigenspace attached to the eigenvalue 1. The said basis is, up to scale, the first 2^{n-1} columns of that matrix. The last assertion follows by standard properties of symmetric real matrices. □

Proof. (of Theorem 11) By Lemma 12, we need to solve for X with rational coordinates the system

$$\begin{aligned}(H_{n-1} + 2^{\frac{n}{2}} I_{2^{n-1}})X &= 2^{\frac{n}{2}} Y \\ H_{n-1} X &= 2^{\frac{n}{2}} Z\end{aligned}$$

or, equivalently

$$Z + X = Y$$
$$H_{n-1}X = 2^{\frac{n}{2}}Z$$

The result follows by $H_{n-1}^2 = 2^{n-1}I_{n-1}$. □

As an example, we treat the case $n = 2$. We get $Y = (2z_1 + z_2, z_1)^T$. The condition $y_1 = \pm 1$ forces $z_1 = -z_2$. We have two self-dual bent functions of sign functions $(z_1, z_1, z_1, -z_1)^T$, with $z_1 = \pm 1$. We give an algorithm to generate all self-dual bent functions of degree at most k.

The algorithm SDB(n,k) is defined as follows.

1. Generate all Z in RM$(k, n-1)$.
2. Compute all Y as $Y := Z + \frac{2H_{n-1}}{2^{\frac{n}{2}}}Z$.
3. If $Y \in \{\pm 1\}^{n-1}$ output (Y, Z), else go to the next Z.

It should be noted that compared to brute force exhaustive search, the computational saving is of order 2^R, with

$$R = 2^n - \sum_{j=0}^{k} \binom{n-1}{j} = 2^{n-1} + \sum_{j=0}^{n-k-1} \binom{n-1}{j} \tag{11}$$

The next result shows that there is a one-to-one correspondence between self-dual and anti-self-dual bent functions.

Theorem 13. *Let $n \geq 2$ be an even integer and Z be arbitrary in $\{\pm 1\}^{n-1}$. Define $Y := Z + \frac{2H_{n-1}}{2^{\frac{n}{2}}}Z$. If Y is in $\{\pm 1\}^{n-1}$, then the vector $(Z, -Y)$ is the sign function of a self-dual bent function in n variables.*

Proof. Observe the identity

$$(I_{2^{n-1}} + \frac{2H_{n-1}}{2^{\frac{n}{2}}})(I_{2^{n-1}} - \frac{2H_{n-1}}{2^{\frac{n}{2}}}) = -I_{2^{n-1}}.$$

From this we see that

$$Z = Y' - \frac{2H_{n-1}}{2^{\frac{n}{2}}}Y',$$

where $Y' = -Y$. By the analogue of Theorem 1 for anti-self-dual bent functions, the result follows. □

From this result follows a generation algorithm for anti-self-dual bent functions of degree at most k. We define the algorithm NSDB(n,k) as follows.

1. Generate all Z in $\mathrm{RM}(k, n-1)$.
2. Compute all Y as $Y := Z - \frac{2H_{n-1}}{2^{\frac{n}{2}}}Z$.
3. If $Y \in \{\pm 1\}^{n-1}$, output (Y, Z), else go to the next Z.

Finally, we point out a connection with *plateaued functions* [9]. Recall that a Boolean function f of n variables is plateaued of order r if the entries of $H_n(-1)^f$ have magnitude either zero or $2^{n-\frac{r}{2}}$.

Theorem 14. *Let $n \geq 2$ be an even integer and Z be arbitrary in $\{\pm 1\}^{n-1}$. Define $Y := Z + \frac{2H_{n-1}}{2^{\frac{n}{2}}}Z$. If Y is in $\{\pm 1\}^{n-1}$, then both Y and Z are sign functions of plateaued Boolean functions of order $n-2$ of $n-1$ variables.*

Proof. Observe that the entries of $Y - Z$ take values in the set $\{0, \pm 2\}$, and therefore the entries of Z are in the set $\{0, \pm 2^{\frac{n}{2}}\}$. Similarly, by the proof of the preceding Theorem, $Z := -Y + \frac{2H_{n-1}}{2^{\frac{n}{2}}}Y$. By the same argument as previous, the entries of Y are in the set $\{0, \pm 2^{\frac{n}{2}}\}$. □

6 Classification

The following results were obtained by using the algorithms SDB(n,k) and NSDB(n,k) for $n \leq 6$ and $k \leq \frac{n}{2}$.. We consider the self-dual bent functions f and g to be equivalent when $g(x) = f(Ax + b) + b \cdot x + c$, where $AA^T = I$, $b \in \mathbb{Z}_2^n$, $\mathrm{wt}(b)$ even, and $c \in \mathbb{Z}_2$.

There is only one self-dual bent function in two variables up to complementation: $(1, 1, 1, -1)$ or x_1x_2. The only anti-self-dual bent function in two variables up to complementation is $(1, -1, -1, -1)$.

We have classified all self-dual bent functions of up to 6 variables. Table 1 gives a representative from each equivalence class and the number of functions in each class. An expression like $12 + 34$ denotes $x_1x_2 + x_3x_4$.

We have classified all quadratic self-dual bent functions of 8 variables. Table 2 gives a representative from each equivalence class and the number of functions in each class.

Table 1: *Self-Dual Bent Functions of 4 and 6 Variables*

Representative from equivalence class	Size
12	1
Total number of functions of 2 variables	1
$12 + 34$	12
$12 + 13 + 14 + 23 + 24 + 34 + 1$	8
Total number of functions of 4 variables	20
$12 + 34 + 56$	480
$12 + 34 + 35 + 36 + 45 + 46 + 56 + 3$	240
$12 + 13 + 14 + 15 + 16 + 23 + 24 + 25 + 26 + 34 + 35 + 36 + 45 + 46 + 56 + 1 + 2$	32
$134 + 234 + 156 + 256 + 12 + 35 + 46 + 56$	11 520
$126 + 136 + 125 + 135 + 246 + 346 + 245 + 345 + 12 + 15 + 26 + 34 + 36 + 45 + 56$	5760
$126 + 136 + 145 + 135 + 246 + 236 + 245 + 345 + 12 + 15 + 25 + 34 + 36 + 46 + 56$	23 040
$456 + 356 + 145 + 246 + 135 + 236 + 124 + 123 + 15 + 26 + 34 + 35 + 36 + 45 + 46 + 3$	1440
$123 + 124 + 134 + 126 + 125 + 136 + 135 + 234 + 236 + 235 + 146 + 145 + 156 + 246 + 245 + 346 + 345 + 256 + 356 + 456 + 14 + 25 + 36 + 45 + 46 + 56 + 1 + 2 + 3$	384
Total number of functions of 6 variables	42 896

Table 2: *Quadratic Self-Dual Bent Functions of 8 Variables*

Representative from equivalence class	Size
$12 + 34 + 56 + 78$	30 720
$12 + 34 + 56 + 57 + 58 + 67 + 68 + 78 + 5$	15 360
$13 + 14 + 15 + 26 + 27 + 28 + 34 + 35 + 45 + 67 + 68 + 78 + 1 + 2$	2048
Number of quadratic functions of 8 variables	48 128

7 Conclusions

In this work we have explored the class of self-dual bent functions and characterized it by the Rayleigh quotient of the Hadamard matrix of Sylvester type. It would be interesting to obtain lower bounds on the Rayleigh quotient of Boolean functions in an odd number of variables. We have determined all self-dual bent functions of at most 6 variables and all quadratic self-dual bent functions of 8 variables. In general, characterizing the class of quadratic self-dual bent functions is a difficult problem. The open question is to determine whether there is more than the Maiorana-McFarland type of subsection 4.1. We have also given some symmetries that preserve the self-dual class in subsection 4.2. It would be interesting to know whether there are no more symmetries. More connections with the theory of self-dual binary codes, for instance weight enumerators, is a goal worth pursuing.

References

[1] Dobbertin, H.: Construction of bent functions and balanced Boolean functions with high nonlinearity. In *Fast Software Encryption*, *Lecture Notes in Comput. Sci.*, vol. 1008, pp. 61–74. Springer, Berlin, 1995.

[2] Dillon, J. F.: *Elementary Hadamard Difference Sets*. Ph.D. thesis, Univ. Maryland, 1974.

[3] Demmel, J. W.: *Applied Numerical Linear Algebra*. SIAM, Philadelphia, PA, 1997.

[4] Carlet, C.: Boolean functions for cryptography and error correcting codes. To appear in Boolean Methods and Models, Cambridge University Press.

[5] Danielsen, L. E., Parker, M. G.: Edge local complementation and equivalence of binary linear codes, 2008. To appear in Des. Codes Cryptogr. arXiv:0710.2243.

[6] Janusz, G. J.: Parametrization of self-dual codes by orthogonal matrices. *Finite Fields Appl.* 13(3), 450–491, 2007.

[7] Riera, C., Parker, M. G.: Generalised bent criteria for Boolean functions (I). *IEEE Trans Inform. Theory* 52(9), 4142–4159, 2006. arXiv:cs.IT/0502049.

[8] Carlet, C.: On the secondary constructions of resilient and bent functions. In *Coding, Cryptography and Combinatorics*, *Progr. Comput. Sci. Appl. Logic*, vol. 23, pp. 3–28. Birkhäuser, Basel, 2004.

[9] Zheng, Y., Zhang, X.-M.: On plateaued functions. *IEEE Trans. Inform. Theory* 47(3), 1215–1223, 2001.