Perfect Quaternion Sequences and Quaternion-type Hadamard Matrices

Aidan Bennett

Abstract

This report details an algorithm to exhaustively enumerate perfect quaternion sequences and quaternionic Hadamard matrices over a particular alphabet of quaternions. The previous best enumeration was able to reach sequences and matrices of sizes as high as 13, whereas we have been able to reach sizes up to and including 21, a significant improvement over the state of the art.

1 Introduction

In this report we exploit a correspondence between a subclass of Williamson-type sequences with perfect sequences over the quaternions, in order to exhaustively enumerate these sequences, as well as Hadamard matrices over the quaternions, and Quaternion-type Hadamard matrices. We develop new techniques to reduce the search space and filter out candidate sequences, before presenting our main algorithm. We then present further optimizations and improvements to the algorithm, as well as some new results.

2 Background

2.1 Quaternions

Sir William Rowan Hamilton is credited with first introducing quaternions, and had initially thought of them as a way of modelling three dimensional space algebraically, in the same way that the complex numbers are used for two dimensional space [5].

Definition 1. *The algebra of* quaternions *is defined as*

$$\mathbb{H} := \{\alpha + \beta i + \gamma j + \delta k \mid \alpha, \beta, \gamma, \delta \in \mathbb{R}\}$$

where i, j, and k are the basic unit quaternions*, satisfying $i^2 = j^2 = -1, ij = k, ji = -k$.*

It is important to note that due to the aforementioned conditions, the quaternions are not commutative. It is fairly straightforward to verify that the quaternions form a skew field. For the following definitions, let $w = \alpha + \beta i + \gamma j + \delta k \in \mathbb{H}$. Analogous to the complex numbers, we call α the *real part* of the quaternion, and $\beta i + \gamma j + \delta k$ is the *imaginary part*. The quaternions are equipped with a conjugation operation *, where $(\alpha + \beta i + \gamma j + \delta k)^* = (\alpha - \beta i - \gamma j - \delta k)$. It can be easily seen that it satisfies

$$(q_1 + q_2)^* = q_1^* + q_2^* \quad \text{and} \quad (q_1 q_2)^* = q_2^* q_1^*$$

The norm defined on $\mathbb{H}$ is also defined similarly to complex numbers:

$$||w|| := \sqrt{\alpha^2 + \beta^2 + \gamma^2 + \delta^2} = \sqrt{ww^*} = \sqrt{w^* w}$$

We define $q := \dfrac{1 + i + j + k}{2}$, a unit quaternion satisfying $q^3 = -1$. We use this to define some sets we use in this report:

$$\mathcal{Q}_8 := \{\pm 1, \pm i, \pm j, \pm k\}, \quad \mathcal{Q}_+ := \mathcal{Q}_8 \cup q\mathcal{Q}_8, \quad \mathcal{Q}_{24} := \mathcal{Q}_+ \cup q^2 \mathcal{Q}_8$$

$\mathcal{Q}_8$ is a group under multiplication, as is $\mathcal{Q}_{24}$. However, $\mathcal{Q}_+$ is not, as it is not closed (i.e., $q^2 = qq \notin \mathcal{Q}_+$).

2.2 Perfect sequences

In this section we provide some necessary definitions and results that we will use regarding perfect sequences, Williamson-type sequences, and quaternion-type sequences. We can measure the correlation between two finite sequences A and B using a *periodic crosscorrelation function.*

Definition 2. *Let $A = [a_0, a_1, \ldots, a_{n-1}]$ and $B = [b_0, b_1, \ldots, b_{n-1}]$ be sequences of length n. The* periodic crosscorrelation *of A and B with offset t is defined as*

$$R_{A,B}(t) := \sum_{r=0}^{n-1} a_r b^*_{r+t \bmod n}$$

It should be clear from the definition, and from the fact that the quaternions do not commute, that we do not in general have $R_{A,B}(t) = R_{B,A}(t)$, for two sequences A and B (when this condition does hold, we say A and B are *pairwise amicable*). It is also useful to define, for a given sequence A, a *periodic autocorrelation function.* However, due again to the noncommutativity of the quaternions, we require separate definitions for left and right autocorrelation.

Definition 3. *Let $A = [a_0, a_1, \ldots, a_{n-1}]$ be a sequence of length n. The* left periodic autocorrelation *and the* right periodic autocorrelation *of A with offset t are defined as*

$$R^L_A(t) := \sum_{r=0}^{n-1} a^*_r a_{r+t \bmod n}, \quad \textit{and} \quad R^R_A(t) := \sum_{r=0}^{n-1} a_r a^*_{r+t \bmod n}$$

respectively.

A sequence is called *left perfect* if its left autocorrelation is zero for all non-trivial shifts, and it is called *right perfect* if its right autocorrelation is zero for all non-trivial shifts, i.e., $1 \leq t < n$. A very useful result, Kuznetsov [6] proved that a sequence of quaternions is left perfect if and only if it is right perfect. Thus, going forward, we shall without loss of generality define the *periodic autocorrelation* of A with shift t to be

$$R_A(t) := R_{A,A}(t) = \sum_{r=0}^{n-1} a_r a^*_{r+t \bmod n}$$

Further, we now call a sequence (real or quaternionic) *perfect* if its autocorrelation is 0 for all non-trivial t.

A sequence consisting of entries strictly in $\{\pm 1\}$ is called a *binary sequence.* A quadruple of binary sequences (A, B, C, D), each of length n, for which $R_{X,Y}(t) = R_{Y,X}(t)$, for any $X, Y \in \{A, B, C, D\}$, and for any t is called a quadruple of *Williamson-type (WT) sequences.* Useful for the following definitions, we denote the *Kronecker delta*:

$$\delta_{0,t} = \begin{cases} 1 & \text{if } t = 0, \\ 0 & \text{otherwise} \end{cases}$$

We are now able to define some of the main objects of interest to this project.

Definition 4. *A quadruple of binary sequences (A, B, C, D), each of length n, is called a quadruple of* quaternion-type (QT) sequences *if it satisfies the following conditions when $0 \leq t < n$:*

$$R_A(t) + R_B(t) + R_C(t) + R_D(t) = \delta_{0,t} 4n, \tag{1}$$
$$R_{A,B}(t) - R_{B,A}(t) + R_{C,D}(t) - R_{D,C}(t) = 0, \tag{2}$$
$$R_{A,C}(t) - R_{C,A}(t) + R_{D,B}(t) - R_{B,D}(t) = 0, \tag{3}$$
$$R_{A,D}(t) - R_{D,A}(t) + R_{B,C}(t) - R_{C,B}(t) = 0. \tag{4}$$

If we replace conditions (2), (3), and (4) with the stronger condition that $R_{X,Y}(t) = R_{Y,X}(t)$, for any $X, Y \in \{A, B, C, D\}$, then (A, B, C, D) are called a quadruple of *Williamson-type (WT) sequences.* If we further require that for each sequence $X = [x_0, x_1, \ldots, x_{n-1}] \in \{A, B, C, D\}$, that $x_i = x_{n-i}$, for $1 \leq i \leq n-1$, then (A, B, C, D) is called a quadruple of *Williamson sequences.*

2.3 Hadamard Matrices

Definition 5. *An $n \times n$ matrix H with entries consisting of only ± 1 is called a* (real) Hadamard matrix *if all of its rows are pairwise orthogonal; that is, $HH^\top = nI_n$*

This definition generalizes in a natural way to Hadamard matrices over quaternions.

Definition 6. *An $n \times n$ matrix H with entries consisting of only unit length quaternions is called a* (quaternionic) Hadamard matrix *if $HH^* = nI_n$, where H^* denotes the conjugate transpose of H.*

A Hadamard matrix (real or quaternionic) is called *circulant* if each of its rows can be obtained from cyclic shifts of the first row. Any perfect sequence consisting of only ± 1 (or in the quaternionic case, unit length quaternions) corresponds directly with a Hadamard matrix, by making the first row to be the perfect sequence, and all other rows to be cyclic shifts of the first row.

One important construction for Hadamard matrices was given by Williamson [10] in 1944. He showed that if A, B, C, D are $n \times n$ pairwise commuting symmetric matrices, satisfying $A^2 + B^2 + C^2 + D^2 = 4I_n$, then the following is a Hadamard matrix.

$$\begin{bmatrix} A & B & C & D \\ -B & A & -D & C \\ -C & D & A & -B \\ -D & -C & B & A \end{bmatrix}$$

Baumert and Hall [3] showed that the conditions could be loosened, to only needing to satisfy the following:

$$\begin{aligned}
AA^\top + BB^\top + CC^\top + DD^\top &= 4nI_n \\
BA^\top - AB^\top + DC^\top - CD^\top &= 0 \\
CA^\top - AC^\top + BD^\top - DB^\top &= 0 \\
BA^\top - AB^\top + DC^\top - CD^\top &= 0
\end{aligned}$$

Quadruples of matrices (A, B, C, D) satisfying the latter conditions are called *quaternion-type matrices*. They can be generated from a quaternion-type sequence (W, X, Y, Z) by letting A be the circulant matrix generated by W, B be the circulant matrix generated by X, etc. Similarly, *Williamson-type* matrices replace the latter four conditions with the condition that $XY^\top = YX^\top$, for each pair $X, Y \in \{A, B, C, D\}$. These are constructed from Williamson-type sequences in the same way.

3 Algorithm for enumerating QT sequences

3.1 Some important results

We require a few more results for our algorithm, which we present here. The following result, proven by Acevedo and Dietrich, demonstrates a connection between QT sequences and perfect quaternionic sequences.

Theorem 7 (cf.[1, Theorem 2.4])**.** *There is a one-to-one correspondence between quadruples of QT sequences, and perfect sequences over $\mathcal{Q}_+$, constructed as follows. Given a QT sequence (A, B, C, D), in which the rth entry of each sequence is denoted (a_r, b_r, c_r, d_r), the rth entry of the corresponding sequence is given below (denoting '1' as '+', and '−1' and '−').*

s_r	1	i	j	k	q	qi	qj	qk
a_r	$-$	$+$	$+$	$+$	$+$	$+$	$+$	$+$
b_r	$-$	$-$	$+$	$-$	$-$	$+$	$+$	$-$
c_r	$-$	$-$	$-$	$+$	$-$	$-$	$+$	$+$
d_r	$-$	$+$	$-$	$-$	$-$	$+$	$-$	$+$

The remaining part of the mapping is constructed by mapping $(-a_r, -b_r, -c_r, -d_r)$ to $-s_r$ whenever a_r, b_r, c_r, d_r is mapped to s_r.

This correspondence reduces the problem of constructing perfect quaternionic sequences, which is hard, to the problem of constructing QT sequences.

To assist in constructing these QT sequences, we require the use of rowsums. Given a sequence $A = [a_0, a_1, \ldots, a_{n-1}]$, we define the *rowsum* of A to be $r_A := \sum_{i=0}^{n-1} a_i$, the sum of the entries in A. The following lemma uses these to aid in the construction of QT sequences.

Lemma 8 (cf. [10, eq. 15])**.** *Let (A, B, C, D) be a quadruple of QT sequences of length n. Then*

$$r_A^2 + r_B^2 + r_C^2 + r_D^2 = 4n$$

This allows us to limit our search to only those sequences satisfying the above rowsum condition.

3.2 Our Algorithm

Using what we have discussed thus far, we are now able to present an algorithm to efficiently exhaustively enumerate perfect quaternionic sequences (see Algorithm 1). To exhaustively enumerate all of the perfect quaternionic sequences of length n, we may first enumerate all of the possible rowsum decompositions according to Lemma 8; that is, ways to write $4n$ in the form $w^2 + x^2 + y^2 + z^2$, for $w, x, y, z \in \mathbb{Z}$.

For each rowsum, we then enumerate all binary sequences that satisfy the rowsum conditions. We break the composition up into pairs, taking the largest rowsum with the smallest rowsum. We could pair the rowsums differently, however, we tested all possible combinations and found this to be the most efficient.

For each pair, and for each t with $1 \leq t < n$, we compute the autocorrelation of both sequences at offset t, and add the two values together. We can then compare this with the autocorrelation of the two sequences in the other pair. Since the autocorrelations must sum to zero (condition (1)), if the sum of the autocorrelations of one pair is not equal to the negation of the sum of the autocorrelations of the other pair, then the two pairs of sequences do not form a valid quadruple of QT sequences. We can do the same with crosscorrelation, comparing the crosscorrelations in one pair with the other to check the relevant condition out of (2), (3), and (4). Any sequences that have not been filtered out at this point are considered potential QT sequences, and at this point we check the remaining crosscorrelation conditions for each of the remaining candidate sequences to obtain our final list of QT sequences.

3.3 QT equivalence, WT equivalence, and Hadamard equivalence

There are several operations that can be applied to QT sequences, WT sequences, and Hadamard matrices that preserve their respective conditions. We use these operations to define notions of equivalence. We first provide a list of operations on quadruples of sequences that preserve the QT conditions.

- NS: (Negate and Swap) Negate a single sequence of A, B, C, D and swap a single pair of sequences of A, B, C, D.
- AN: (Alternating negation) Multiply every other element of A, B, C and D simultaneously by -1 when the length n is even.
- DE: (Decimate) Apply an automorphism of the cyclic group C_n to the indices of each sequence A, B, C, and D simultaneously.
- CS: (Cyclic Shift) Cyclically shift all the entries of A, B, C, and D simultaneously by any amount.
- DH: (Double Half Shift) Shift the entries of two of A, B, C, and D by $n/2$ when the length n is even.

We call two quadruples of QT sequences equivalent if one can be obtained from the other by applying any combination of the above equivalence operations. Defining such equivalence classes allows us to eliminate symmetries in the search space, as seen in section 3.4.

Algorithm 1 Algorithm for exhaustively enumerating QT sequences of a given order n.

$\mathcal{L} := \emptyset$ // Set that will contain the QT sequences found
// Decompose $4n$ as a sum of four nonnegative integer squares
$\mathcal{D} := \{\, (w, x, y, z) \in \mathbb{N}^4 : w^2 + x^2 + y^2 + z^2 = 4n$ and $w \geq x \geq y \geq z \,\}$
for $(w, x, y, z) \in \mathcal{D}$ **do**
  Construct $\mathcal{W}$, $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$, the sets of binary sequences with rowsums equal to w, x, y, z, respectively
  $\mathcal{L}_1 := \mathcal{W} \times \mathcal{Z}$ and $\mathcal{L}_2 := \mathcal{X} \times \mathcal{Y}$
  // Compute autocorrelations of each candidate pair in $\mathcal{L}_1$
  Compute $[R_W(t) + R_Z(t) : 1 \leq t \leq n/2]$ for all $(W, Z) \in \mathcal{L}_1$ with $R_{W,Z}(t) = R_{Z,W}(t)$ for all $1 \leq t \leq n/2$ and store in a list C_1
  // Compute *negated* autocorrelations of each candidate pair in $\mathcal{L}_2$
  Compute $[-R_X(t) - R_Y(t) : 1 \leq t \leq n/2]$ for all $(X, Y) \in \mathcal{L}_2$ with $R_{X,Y}(t) = R_{Y,X}(t)$ for all $1 \leq t \leq n/2$ and store in a list C_2
  // Matching vectors in C_1 and C_2 provide solutions of (1) and (4)
  for each list of values v appearing in both C_1 and C_2 **do**
    Get the set $\mathcal{S}$ of all the quadruples of sequences generating v in both lists C_1 and C_2
    for each quadruple (W, X, Y, Z) in $\mathcal{S}$ **do**
      if (W, X, Y, Z) is a quadruple of QT sequences **then**
        Add (W, X, Y, Z) to $\mathcal{L}$
      end if
    end for
  end for
end for
return $\mathcal{L}$

We define WT sequence equivalence in a similar way. The following conditions preserve the WT sequence conditions.

- SN: (Single Negate) Negate one sequence of A, B, C, or D.
- SS: (Single Swap) Swap one pair of A, B, C, D.
- SH: (Single Half Shift) Shift the entries of one of A, B, C, D by $n/2$ when n is even.

Most of the equivalence operations that apply to QT sequences are the same for WT sequences, with some exceptions. In the case of WT sequences, we can replace NS with SN, and SS, and we can replace DH with SH.

In the case of Hadamard matrices, we say that two Hadamard matrices are equivalent if one can be obtained from the other by negating rows and columns, or by interchanging rows and columns. We call two QT sequences *Hadamard equivalent* if they generate equivalent Hadamard matrices.

This notion generalizes in a straightforward way to QHMs. Two QHMs are said to be equivalent if one can be obtained from the other by multiplying rows on the left by unit quaternions, by multiplying columns on the right by unit quaternions, or by interchanging rows and columns. These definitions give way to another definition. We call a Hadamard matrix (real or quaternionic) *normalized* if its first row and first column consist of only 1's. Notice that every Hadamard matrix is equivalent to a normalized Hadamard matrix.

3.4 Optimizations

We found many ways to improve on this algorithm. In our search, we noticed that all QT sequences we found are also WT sequences. This led us to the following theorem.

Theorem 9. *All QT sequences are WT.*

This drastically speeds up the search, as instead of verifying that a sequence satisfies all three cross-correlation conditions (i.e., (2), (3), and (4)), we can instead only verify the amicability condition. Further, this allows us to filter out sequences early on if they are not pairwise amicable.

In step 2 of Algorithm 1, we can avoid many decompositions. For any given QT sequence, one can see that we can apply the SS and SN equivalence operations to obtain an equivalent WT sequence whose rowsums (w, x, y, z) satisfy $w \geq x \geq y \geq z \geq 0$. As a result, we are able to ignore rowsum decompositions in any other form, drastically cutting down the search space.

Many optimizations for this search use the discrete Fourier transform (DFT). Given a sequence A, we denote the tth value of the DFT of A by $\mathrm{DFT}_A(t)$. We further define the *power spectral density* of A as $\mathrm{PSD}_A(t) := ||\mathrm{DFT}_A(t)||^2$. The following is a well known result [4], and proves useful in reducing the search space of sequences.

Proposition 10. *(A, B, C, D) is a quadruple of QT sequences of length n if and only if*

$$\mathrm{PSD}_A(t) + \mathrm{PSD}_B(t) + \mathrm{PSD}_C(t) + \mathrm{PSD}_D(t) = 4n$$

for all $t \in \mathbb{Z}$.

This allows us to improve our filtering. When constructing the pairs, we can add the PSD of the two sequences in a given pair together. If they sum to greater than $4n$, then we can remove the pair, as the four sequences would necessarily violate the condition in Proposition 10.

Given two sequences X and Y, we can define the *cross power spectral density* of X and Y by $\mathrm{CPSD}_{X,Y}(t) := \mathrm{DFT}_X(t)\,\mathrm{DFT}_Y(t)^*$. We can use the following proposition for an optimization.

Proposition 11. *Consider the following equations.*

$$\mathrm{CPSD}_{A,B}(t) - \mathrm{CPSD}_{B,A}(t) + \mathrm{CPSD}_{C,D}(t) - \mathrm{CPSD}_{D,C}(t) = 0 \tag{5}$$
$$\mathrm{CPSD}_{A,C}(t) - \mathrm{CPSD}_{C,A}(t) + \mathrm{CPSD}_{D,B}(t) - \mathrm{CPSD}_{B,D}(t) = 0 \tag{6}$$
$$\mathrm{CPSD}_{A,D}(t) - \mathrm{CPSD}_{D,A}(t) + \mathrm{CPSD}_{B,C}(t) - \mathrm{CPSD}_{C,B}(t) = 0 \tag{7}$$

Conditions (5)*,* (6)*, and* (7)*, are equivalent to conditions* (2)*,* (3)*, and* (4)*, respectively (for all* $t \in \mathbb{Z}$*).*

As a result, instead of checking the amicability condition $R_{X,Y}(t) = R_{Y,X}(t)$, we can instead check that $\mathrm{CPSD}_{X,Y}(t) = \mathrm{CPSD}_{Y,X}(t)$. Since the DFT of each sequence is already being computed for the PSD filter, we may cache those values to compute the CPSD values very quickly. In practice, this is much faster than computing the crosscorrelations.

3.5 Equivalence filtering

After applying Algorithm 1, we perform a post processing step to reduce the sequences to canonical representatives of their equivalence classes, via QT equivalence, WT equivalence, and Hadamard equivalence. During our search, we noticed a pattern that led to the following theorem.

Theorem 12. *QT sequences that are QT equivalent are also Hadamard equivalent.*

Thus, to find the list of QT sequences up to Hadamard equivalence, we can first find the list of QT sequences up to QT equivalence. The full pipeline for reducing to equivalence is as follows.

1. Apply Algorithm 1 to obtain a full list of QT sequences up to WT equivalence.

2. Filter out sequences that are WT equivalent by constructing a canonical representative for each WT equivalence class. This representative is done by taking the lexicographic minimum of all WT sequences that can be generated by the equivalence operations {SS, SN, AN, DE, CS, SH}. In practice, this is done by adapting Algorithm 5 from a similar search done by Lumsden, Kotsireas, and Bright [7].

3. Since we now have all WT equivalence classes, but not all QT equivalence classes, we can get the missing QT equivalence classes by doing the following. For each QT sequence (W, X, Y, Z), we add the sequence $(-W, X, Y, Z)$. Afterwards if n is even, for each QT sequence (W, X, Y, Z) we add $(s^{\frac{n}{2}}(W), X, Y, Z)$, where $s^{\frac{n}{2}}(W)$ denotes the SH operation being applied on W.

4. We now filter up to QT equivalence using the same method as in part 2, using the equivalences {NS, AN, DE, CS, DH}, to remove any duplicate sequences up to QT equivalence.

5. We now use the list of sequences up to QT equivalence to create the list of sequences up to Hadamard equivalence. For this, we convert each QT sequence into its corresponding Hadamard matrix, and convert each matrix into a graph representation as described by McKay [8]. We then reduce the graphs using to the canonical forms of their isomorphism classes using Nauty [9]. Two Hadamard matrices are Hadamard equivalent if and only if their graph representations are isomorphic, so we use Nauty to reduce the graph representations up to isomorphism.

3.6 Results

The algorithm was programmed with Rust 1.91.0, with Bash shell scripts to assist in execution. It was run on an AMD EPYC CPU at 2.7 GHz, limited to 4 GiB of RAM. The GNU sort function was used to sort files containing PSD/CPSD data. The program was able to exhaustively enumerate all QT sequences up to and including length 21 in roughly 47 hours. This is a significant improvement over the previous best of 13 [2], which took about a week of computation time. The code, as well as the results of the computation can be found at `https://github.com/colinotp/quaternion-sequences`. Some runtime data can also be found in Table 1.

Table 1: Table of results for Algorithm 1 by length n. W_{equ}, Q_{equ}, and H_{equ} indicate the total number of QT sequences found up to Williamson-type, QT, and Hadamard equivalence, respectively. "Time" is the total enumeration runtime in seconds, "Pairs" is the total number of pairs generated in step 5 (generating these is the bottleneck of the algorithm), and "Space" is the amount of disk space used in MiB.

n	W_{equ}	Q_{equ}	H_{equ}	Time (s)	Pairs	Space
1	1	1	1	0.16	2	0.0
2	1	1	1	0.08	4	0.0
3	1	1	1	0.08	6	0.0
4	2	3	2	0.19	46	0.0
5	1	1	1	0.08	20	0.0
6	1	1	1	0.16	48	0.0
7	2	3	3	0.15	182	0.0
8	3	4	3	0.08	384	0.0
9	4	7	7	0.31	999	0.0
10	2	4	2	0.23	770	0.0
11	1	2	2	0.15	715	0.0
12	5	10	6	0.80	6288	0.2
13	4	6	6	3.14	8216	0.4
14	5	12	10	3.15	5334	0.3
15	4	7	7	23.46	15732	1.0
16	18	44	19	121.20	206480	7.9
17	4	5	5	361.36	39372	3.6
18	28	88	82	2001.79	275184	17.5
19	6	11	11	7753.85	200526	22.0
20	24	84	54	17632.48	625896	46.1
21	7	13	13	141267.63	561519	86.1

References

[1] Santiago Barrera Acevedo and Heiko Dietrich. New infinite families of Williamson Hadamard matrices. *Australasian Journal of Combinatorics*, 73(1):207–219, January 2019.

[2] Santiago Barrera Acevedo, Padraig Ó. Catháin, and Heiko Dietrich. Constructing cocyclic Hadamard matrices of order $4p$. *Journal of Combinatorial Designs*, 27(11):627–642, August 2019.

[3] L. D. Baumert and Marshall Hall. Hadamard matrices of the Williamson type. *Mathematics of Computation*, 19(91):442–447, 1965.

[4] Curtis Bright, Ilias Kotsireas, and Vijay Ganesh. Applying computer algebra systems with SAT solvers to the Williamson conjecture. *Journal of Symbolic Computation*, 100:187–209, September 2020.

[5] William Rowan Hamilton. On quaternions; or on a new system of imaginaries in algebra. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 29(191):26–31, Jul 1846.

[6] Oleg Kuznetsov. *Perfect Sequences over the Real Quaternions*. PhD thesis, Monash University, 2010.

[7] Tyler Lumsden, Ilias Kotsireas, and Curtis Bright. New results on periodic Golay pairs. *Mathematics of Computation*, April 2025.

[8] Brendan D. McKay. Hadamard equivalence via graph isomorphism. *Discrete Mathematics*, 27(2):213–214, 1979.

[9] Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, II. *Journal of Symbolic Computation*, 60:94–112, January 2014.

[10] John Williamson. Hadamard's determinant theorem and the sum of four squares. *Duke Mathematical Journal*, 11(1), March 1944.