Highway to Hull: An Algorithm for Solving the General Matrix Code Equivalence Problem

Alain Couvreur[1,2] and Christophe Levrat[1,2]

[1] Inria
[2] Laboratoire LIX,
École Polytechnique,
Institut Polytechnique de Paris,
France
{alain.couvreur, christophe.levrat}@inria.fr

Abstract. The matrix code equivalence problem consists, given two matrix spaces $\mathcal{C}, \mathcal{D} \subset \mathbb{F}_q^{m \times n}$ of dimension k, in finding invertible matrices $P \in \mathrm{GL}_m(\mathbb{F}_q)$ and $Q \in \mathrm{GL}_n(\mathbb{F}_q)$ such that $\mathcal{D} = P\mathcal{C}Q^{-1}$. Recent signature schemes such as MEDS and ALTEQ relate their security to the hardness of this problem. Naranayan *et. al.* recently published an algorithm solving this problem in the case $k = n = m$ in $\widetilde{\mathcal{O}}(q^{\frac{k}{2}})$ operations. We present a different algorithm which solves the problem in the general case. Our approach consists in reducing the problem to the matrix code conjugacy problem, *i.e.* the case $P = Q$. For the latter problem, similarly to the permutation code equivalence problem in Hamming metric, a natural invariant based on the *Hull* of the code can be used. Next, the equivalence of codes can be deduced using a usual list collision argument. For $k = m = n$, our algorithm achieves the same complexity as in the aforementioned reference. However, it extends to a much broader range of parameters.

Introduction

In the last decades so-called *equivalence problems* have frequently been used for cryptographic applications. The first examples probably come from multivariate cryptography with Matsumoto Imai [19] or HFE [22] schemes whose security relies on the hardness of the polynomial isomorphism problem: deciding whether two spaces of polynomials are equivalent with respect to a linear or affine change of variables.

In the recent years, we observed an intensification of this trend but also a diversification of the equivalence problems used for cryptography. In particular, in NIST's recent on-ramp call for signature[3], many signature schemes involve equivalence problems which are not the polynomial isomorphism one. For instance, *Hawk*'s [5] security rests among others on the hardness of the Lattice

[3] https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures

Isomorphism Problem (LIP), LESS [1] rests on the monomial equivalence of Hamming metric codes, MEDS [9] on the matrix code equivalence and ALTEQ [4] on the equivalence of alternate trilinear forms. The latter problem (equivalence of alternate trilinear forms) is actually a sub-case of the former: the matrix code equivalence problem, which is the purpose of the present article. The other way around the two problems are proved to be polynomially equivalent [14, Prop. 8.3].

Given two matrix spaces $\mathcal{C}, \mathcal{D} \subset \mathbb{F}_q^{m\times n}$, the *matrix code equivalence problem* consists in deciding whether there exists $P \in \mathrm{GL}_m(\mathbb{F}_q)$ and $Q \in \mathrm{GL}_n(\mathbb{F}_q)$ such that $\mathcal{C} = P\mathcal{D}Q^{-1}$. The search version of the problem asks to return, if exists, a pair P, Q providing the equivalence. If its use for cryptography is rather new, this problem is known for a long time in algebraic complexity theory where it is usually formulated in an equivalent way as the 3–*tensor isomorphism* problem. This problem is assumed to be hard and is in particular known to be at least as hard as the monomial code equivalence problem (see [14] or [11]).

Our contribution

In this article, we propose a new algorithm for solving the matrix code equivalence problem, or equivalently the 3–tensor isomorphism problem. Given equivalent k–dimensional $m \times n$ matrix spaces with entries in $\mathbb{F}_q$, we are able to solve the search equivalence problem in a complexity $\widetilde{\mathcal{O}}(q^{s-\frac{m-1}{2}})$ where $s = \min\{k, mn-k\}$. Note in particular that in the specific case $k = m = n$ which is the one that is used in the parameters of MEDS and ALTEQ, we achieve the complexity $\widetilde{\mathcal{O}}(q^{\frac{k}{2}})$ which is the one achieved by Narayanan, Qiao and Tang [20]. However,

1. our algorithm rests on completely different invariants;
2. our result does not require the parameters k, m, n to be equal, while this is necessary for Narayanan *et. al.*'s algorithm to run.
   Note that, if the equivalence of alternate trilinear forms problem on which ALTEQ is built requires by design to have $k = m = n$, there is no need to instantiate MEDS with such a constraint on k, m, n. It turns out that MEDS' proposed parameters [9] satisfy this condition making them vulnerable to Narayanan *et. al.*'s attack, but MEDS' designers could have easily circumvented the aforementioned attack just by breaking the symmetry on the parameters k, m, n. Still, the algorithm introduced in the present article attacks a much broader range of triples k, m, n.

A specificity of our algorithm is that, taking its inspiration from the Hamming metric counterpart of the code equivalence problem problem and Sendrier's famous *support splitting algorithm* [26], we use the *Hull* of the code, *i.e.* its intersection with its orthogonal space w.r.t some given bilinear form.

Related works

MEDS and ALTEQ schemes [9, 4] were both submitted to NIST's on-ramp call for digital signatures. Before, ALTEQ's and MEDS' specifications were respectively presented in the articles [29] and [10]. In [3], Beullens proposes a new algorithm solving the trilinear form equivalence problem, harming the proposed parameters for ALTEQ. More recently, Narayanan, Qiao and Tang [20] proposed an algorithm solving the same problem but also the matrix code equivalence problem in the case of k–dimensional spaces of $k \times k$ matrices. Their approach combines a collision list argument with a nice algebraic invariant and achieves a complexity in $\widetilde{\mathcal{O}}(q^{\frac{k}{2}})$.

Outline of the article

In Section 1, we introduce the matrix code equivalence problem as well as some related problems and algorithms solving them. In Section 2, we first state some key observations at the heart of our algorithm, then give a brief presentation of the algorithm as well as its complexity. The algorithm itself can be divided into three steps. The crucial one is the reduction to a specific instance of the matrix code conjugacy problem, where conjugate one-dimensional subspaces inside the two codes are given. This reduction is presented in Section 3. We then solve the conjugacy problem in question using an algorithm presented in Section 4. Finally, we explain in Section 5 how to deduce a solution to the initial matrix code equivalence problem from the solution to the conjugacy problem found in the previous step.

Acknowledgements

The second author is funded by Inria and the French Cybersecurity Agency (ANSSI). This work benefited from the financial support of the French government and the *Agence Nationale de la Recherche (ANR)* through the *Plan France 2030* via the project ANR-22-PETQ-0008. The authors are part of a collaborative research project *Barracuda* with reference ANR-21-CE39-0009-BARRACUDA. The first author is partially funded by Horizon-Europe MSCA-DN project *Encode*.

1 The matrix code equivalence problem

Definition 1 (Matrix code equivalence problem). *Let m, n, k be positive integers. Consider two k-dimensional linear subspaces $\mathcal{C}, \mathcal{D}$ of $\mathbb{F}_q^{m\times n}$. The matrix code equivalence problem $\mathsf{MCE}_{m,n,k}(\mathcal{C}, \mathcal{D})$ consists in finding (if exist) matrices $P \in \mathrm{GL}_m(\mathbb{F}_q), Q \in \mathrm{GL}_n(\mathbb{F}_q)$ such that*

$$\mathcal{D} = P\mathcal{C}Q^{-1}.$$

When $m = n = k$, we call it the cubic matrix code equivalence problem*: $\mathsf{CMCE}_n(\mathcal{C}, \mathcal{D})$.*

Remark 1. We may suppose that $m \leqslant n$. Indeed, if $\mathcal{D} = P\mathcal{C}Q^{-1}$ then

$$\mathcal{D}^\top = (Q^{-1})^\top \mathcal{C}^\top P^\top$$

so any algorithm solving the case $m \leqslant n$ can also be used, after transposing the whole problem, to solve the case $n \leqslant m$. In the remainder of this article, we will always suppose that $m \leqslant n$.

The CMCE problem is notably the basis of the former NIST signature scheme candidate MEDS. A polynomial-time equivalent problem [24], the *alternating trilinear form equivalence problem*, underpins the former NIST signature candidate ALTEQ. An attack against these problems was recently described by Naranayan, Qiao and Tang in [20].

1.1 Related problems

The trilinear forms equivalence problem

Definition 2 (Trilinear Form Equivalence Problem (TFE)). *The* trilinear forms equivalence *problem* $\mathsf{TFE}_{m,n,k}$ *is the following. Given two trilinear forms* $f, g\colon \mathbb{F}_q^m \times \mathbb{F}_q^m \times \mathbb{F}_q^k \to \mathbb{F}_q$*, find three matrices* $(P, Q, R) \in \mathrm{GL}_m(\mathbb{F}_q) \times \mathrm{GL}_n(\mathbb{F}_q) \times \mathrm{GL}_k(\mathbb{F}_q)$ *such that for any* $x, y, z \in \mathbb{F}_q^m \times \mathbb{F}_q^n \times \mathbb{F}_q^k$*,*

$$(Px, Qy, Rz) = g(x, y, z).$$

Lemma 1. *The problem* $\mathsf{MCE}_{m,n,k}$ *admits a (deterministic) polynomial-time reduction to* $\mathsf{TFE}_{m,n,k}$*.*

Proof. Let $(\mathcal{C}, \mathcal{D})$ be an instance of $\mathsf{MCE}_{m,n,k}$. Denote by $(C_1, \ldots, C_k)$ a basis of $\mathcal{C}$ and by $(D_1, \ldots, D_k)$ a basis of $\mathcal{D}$. We may define the trilinear forms

$$f\colon (x, y, z) \mapsto \sum_{i,j,\ell} (C_k)_{ij} x_i y_j z_\ell$$
$$g\colon (x, y, z) \mapsto \sum_{i,j,\ell} (D_k)_{ij} x_i y_j z_\ell.$$

If $g(x, y, z) = f(Px, Qy, Rz)$ for all $(x, y, z) \in \mathbb{F}_q^m \times \mathbb{F}_q^n \times \mathbb{F}_q^k$, then straightforward computations show that $\mathcal{D} = P^\top \mathcal{C} Q^\top$, and the element R_{ij} is the i-th coordinate of D_j when expressed in the basis $(P^\top C_1 Q^\top, \ldots, P^\top C_k Q^\top)$ of $\mathcal{D}$. □

Remark 2. Even if the two aforementioned problems are actually polynomially equivalent (see [14]), the converse of this construction does not directly yield a deterministic polynomial-time reduction of $\mathsf{TFE}_{m,n,k}$ to $\mathsf{MCE}_{m,n,k}$. Indeed, let (f, g) be an instance of $\mathsf{TFE}_{m,n,k}$. We may write

$$f\colon (x, y, z) \mapsto \sum_{i,j,k} c_{ijk} x_i y_j z_k$$
$$g\colon (x, y, z) \mapsto \sum_{i,j,k} d_{ijk} x_i y_j z_k.$$

For $r \in \{1 \dots k\}$, construct the matrices $C_r = (c_{ijr})_{i,j}$ and $D_r = (d_{ijr})_{i,j}$. Now, it is not guaranteed that the codes $\mathcal{C} = \mathrm{Span}(C_1, \dots, C_k)$ and $\mathcal{D} = \mathrm{Span}(D_1, \dots, D_k)$ are k-dimensional. However, if they are, this is indeed an instance of $\mathsf{MCE}_{m,n,k}$ and if we find a solution (P, Q) such that $\mathcal{D} = P\mathcal{C}Q$, we can immediately solve this instance of $\mathsf{TFE}_{m,n,k}$. Indeed, consider the matrix $R = (r_{ij}) \in \mathrm{GL}_k(\mathbb{F}_q)$ where r_{ij} is i-th coordinate of D_j when expressed in the basis $(PC_1Q, \dots, PC_kQ)$ of $\mathcal{D}$. Then, we have $f(P^\top x, Q^\top y, Rz) = g(x, y, z)$ for all (x, y, z). In practice, given a random trilinear form, the matrices C_i are random elements of $\mathbb{F}_q^{m\times n}$, and they are very likely to be linearly independent.

Alternate trilinear form equivalence problem A sub-case of the trilinear form equivalence problem that has been considered for the design of ALTEQ is the *alternate trilinear forms equivalence problem* ATFE. An *alternate* trilinear form is a trilinear form $f : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n$ such that for any $x \in \mathbb{F}_q^n$,

$$f(x, x, \cdot) \equiv f(x, \cdot, x) \equiv f(\cdot, x, x) \equiv 0.$$

This is equivalent to the fact that, given any permutation $\sigma \in \mathfrak{S}_3$ (the group of permutation on 3 letters), and any triple $(x_1, x_2, x_3) \in (\mathbb{F}_q^n)^3$,

$$f(x_{\sigma(1)}, x_{\sigma(2)}, x_{\sigma(3)}) = \varepsilon(\sigma) f(x_1, x_2, x_3),$$

where $\varepsilon(\sigma)$ denotes the signature of the permutation σ. This definition leads to the following problem.

Definition 3 (Alternate Trilinear Form Equivalence Problem (ATFE)). *The alternate trilinear forms equivalence problem* $\mathsf{TFE}_{m,n,k}$ *is the following. Given two alternate trilinear forms* $f, g \colon \mathbb{F}_q^m \times \mathbb{F}_q^m \times \mathbb{F}_q^k \to \mathbb{F}_q$, *find a matrix* $P \in \mathrm{GL}_n(\mathbb{F}_q)$ *such that for any* $x, y, z \in \mathbb{F}_q^m \times \mathbb{F}_q^n \times \mathbb{F}_q^k$,

$$f(Px, Py, Pz) = g(x, y, z).$$

3–tensor isomorphism. On the tensor product $\mathbb{F}_q^m \otimes \mathbb{F}_q^n \otimes \mathbb{F}_q^k$, there is a natural action of $\mathrm{GL}_m(\mathbb{F}_q) \times \mathrm{GL}_n(\mathbb{F}_q) \times \mathrm{GL}_k(\mathbb{F}_q)$, and the 3–tensor isomorphism problem consists in deciding whether two tensors $T_1, T_2 \in \mathbb{F}_q^m \otimes \mathbb{F}_q^n \otimes \mathbb{F}_q^k$ are in the same orbit with respect to the aforementioned group action.

The equivalence between the matrix code equivalence problem and the 3–tensor isomorphism one, is very explicit. Given two tensors, one can consider the matrix subspaces of $\mathbb{F}_q^m \times \mathbb{F}_q^n$ spanned by their "slices" and the tensors are isomorphic if and only if the corresponding matrix spaces are equivalent. Conversely, given two matrix spaces, one can take a basis for each one, and stack elements of a basis in order to create a 3–tensor. Then, the matrix spaces will be equivalent if and only if the corresponding 3–tensors are isomorphic.

Remark 3. Note that the terminology of *cubic matrix code equivalence problem* introduced in Definition 1 refers to the corresponding tensors, which will be $n \times n \times n$, *i.e. cubic tensors.*

Similarly, the equivalence between the 3–tensor isomorphism and the equivalence of trilinear forms, can be made explicit since a trilinear form is encoded by a 3-tensor $T \in \mathbb{F}_q^m \otimes \mathbb{F}_q^n \otimes \mathbb{F}_q^k$. The equivalence of the problems mentioned in this section and more is summarized in [14, Figure 2].

Finally, ATFE can be reformulated in terms of the equivalence of *alternate tensors* which are tensors $T \in \mathbb{F}_q^n \otimes \mathbb{F}_q^n \otimes \mathbb{F}_q^n$ such that for any $\sigma \in \mathfrak{S}_3$,

$$\sigma(T) = \varepsilon(\sigma) \cdot T,$$

where $\sigma(T)$ denotes the image of T under the natural action of $\mathfrak{S}_3$ on such 3–tensors and $\varepsilon(\sigma)$ denotes the signature of σ.

Remark 4. Rewriting an instance of MCE as an instance of TFE shows that the problem is symmetric in the three parameters m, n, k. In particular, we may choose to permute m, n, k as we like in our algorithm in order to minimize its complexity. Moreover, as we will explain in Lemma 2, we may also switch k for $mn - k$.

1.2 Related works on attacks on MCE and ATFE

About ALTEQ. In article [29] in which the design of ALTEQ is established, various cryptanalysis techniques are considered to solve ATFE problem. They include algebraic attacks: computing the matrix P as the solution of a quadratic system; or MinRank based attacks (rediscussed further as "Leon–like techniques"). The authors then consider a collision search attack with a cost $\widetilde{\mathcal{O}}(q^{\frac{2n}{3}})$, which they claim to be the best possible. Finally, NIST proposal ALTEQ [4] selects parameters with respect to a finer analysis of algebraic and MinRank based attacks.

About MEDS. For the design of MEDS [10], the authors consider a graph-search based approach inspired from the works of Bouillaguet, Fouque and Véber [6] on the polynomial isomorphism problem for spaces of quadratic forms. This approach leads to an attack of complexity $\widetilde{\mathcal{O}}(q^{\frac{2}{3}(m+n)})$. Also, they consider the possibility of algebraic modelling which turns out to be harder than for ATFE since the unknown correspond to a pair of matrices (P, Q) instead of a single one. They also study a "Leon–like" approach, a reference to Leon's algorithm [17] for determining code equivalence that consists in harvesting minimum weight codewords to determine the code equivalence. When transposed to the matrix code setting, the Hamming weight is replaced by the rank and such approach is nothing but the aforementioned MinRank based technique, which, following a recent result from Beullens [2] on Hamming metric code equivalence, can be combined with a collision search technique. MEDS' parameter selection rests on the complexity of both algebraic attacks and Leon–like ones.

Subsequent attacks. Recently, two attacks taking their inspiration from the Graph–based techniques of Bouillaguet, Fouque and Véber [6] appeared in the literature.

Beullens' attack. First, Beullens proposed a graph–search–based technique to solve ATFE. His attack turns out to be particularly efficient for low values of n. For instance, for n odd, he could achieve a complexity in $\mathcal{O}(q^{(n-5)/2}n^{11}+q^{n-7}n^6)$. This permitted to identify weak keys in [29].

Narayanan, Qiao and Tang's attack. In [20], the authors introduced a new algorithm solving both ATFE and MCE in the specific case $k = m = n$. We conclude this section by sketching the principle of this algorithm in order to point out the need for being in the cubic case $k = m = n$.

As already explained in § 1.1, the problem can be reformulated into that of the equivalence of two trilinear forms

$$f : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \longrightarrow \mathbb{F}_q \quad \text{and} \quad g : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \longrightarrow \mathbb{F}_q,$$

where we look for a triple $P, Q, R \in \mathrm{GL}_n(\mathbb{F}_q)$ such that for any $x, y, z \in \mathbb{F}_q^n$, $g(x, y, z) = f(Px, Qy, Rz)$.

The idea of the algorithm consists first in guessing a pair $(x_1, x_1') \in (\mathbb{F}_q^n)^2$ such that $x_1' = Px_1$ and $f(x_1, \cdot, \cdot)$ is a bilinear form of rank $n-1$. Next, due to the rank constraint, from x_1 can be deduced a unique y_1 (up to scalar multiplication) such that $f(x_1, y_1, \cdot) \equiv 0$. Similarly, they deduce a z_1 such that $f(\cdot, y_1, z_1) \equiv 0$, and an x_2 such that $f(x_2, \cdot, z_1) \equiv 0$ and so on. By this manner, they construct 3 sequences $x_1, \ldots, x_n$, $y_1, \ldots, y_n$ and $z_1, \ldots, z_n$ for f and similarly construct $x_1', \cdots, x_n'$, $y_1', \ldots, y_n'$ and $z_1', \ldots, z_n'$ for f'. Stacking these vectors as columns of $n \times n$ matrices, we get 6 matrices X, Y, Z, X', Y' and Z' that will be invertible with a high probability.

The key observation is that

$$X' = PX \quad Y' = QY \quad \text{and} \quad Z' = RZ.$$

Therefore, for any x, y, z

$$f(X'x, Y'y, Z'z) = f(PXx, QYy, QZz) = g(Xx, Yy, Zz).$$

Thus, (up to some action of diagonal matrices that we do not discuss here) the trilinear forms $f_{x_1} \stackrel{\text{def}}{=} f(X' \cdot, Y' \cdot, Z' \cdot)$ and $g_{x_1'} \stackrel{\text{def}}{=} g(X \cdot, Y \cdot, Z \cdot)$ coincide.

In view of this observation, the algorithm solving MCE consists in a collision search between two dictionaries. The first one collects pairs (f_{x_1}, x_1), the left–hand term being the search key and the right–hand one being the corresponding value, and the second one collects pairs $(g_{x_1'}, x_1')$. Once such a collision is found, which requires $O(q^{n/2})$ trials, determining the equivalence becomes easy (see [20] for further details).

Conclusion about Narayanan, Qiao and Tang. It should be emphasized that the crux of their algorithm rests on the unique possibility (up to scalar multiplication) of passing from x_i to y_i, from y_i to z_i and from z_i to x_{i+1}. Such a technique is possible only because at each step, the corresponding bilinear form is represented by a rank $n-1$ matrix of size $n \times n$. Hence, their approach strongly rests on the fact that all the involved matrices have the same sizes, *i.e.* that they lie in the cubic case $k = m = n$.

2 Technical overview

In this article, we propose an algorithm to solve $\mathsf{MCE}_{m,n,k}(\mathcal{C},\mathcal{D})$ for the range of parameters m,n,k such that $n \geqslant m$ and

$$k < m^2 - 1 \text{ or } mn - k < m^2 - 1$$

(see Remark 6). This includes the cubic case, *i.e.*, $k = m = n$, in which its complexity turns out to be similar to that of [20].

2.1 Preliminaries

Our algorithm will use in a crucial way the notion of dual matrix code and that of *Hull*. We give both definitions below.

Definition 4. *Let $\mathcal{C} \subset \mathbb{F}_q^{m\times n}$ be a linear code. The* dual *of $\mathcal{C}$ is the code*

$$\mathcal{C}^\perp \stackrel{def}{=} \{M \in \mathbb{F}_q^{m\times n} \mid \forall C \in \mathcal{C}, \mathrm{Tr}(M^\top C) = 0\}.$$

Definition 5. *Let $\mathcal{C} \subset \mathbb{F}_q^{m\times m}$ be a matrix code. We will call* hull *of $\mathcal{C}$ the code*

$$h(C) = \{M \in \mathcal{C} \mid \forall C \in \mathcal{C}, \mathrm{Tr}(MC) = 0\}.$$

Remark 5. Beware that the hull is **not** the intersection of $\mathcal{C}$ with its dual as defined in Definition 4. It is the intersection with another orthogonal subspace, this time with respect to the bilinear form

$$(X,Y) \mapsto \mathrm{Tr}(XY).$$

The definition of the hull (Defintion 5) is the only place of the article where this nonstandard bilinear form is used. Besides, every dual or orthogonal complement which appears in the article is taken with respect to the usual inner product

$$(X,Y) \mapsto \mathrm{Tr}(X^\top Y).$$

The subsequent lemmas yield two key observations for our algorithms are the following statements which claim that:

1. if two codes are equivalent, so are their duals;
2. if two codes are conjugate, so are their hulls.

Lemma 2. *Let $\mathcal{C},\mathcal{D} \subset \mathbb{F}_q^{m\times n}$ be two $\mathbb{F}_q$-vector spaces, and $P \in \mathrm{GL}_m(\mathbb{F}_q)$, $Q \in \mathrm{GL}_n(\mathbb{F}_q)$ be matrices such that $\mathcal{D} = P\mathcal{C}Q^{-1}$. Then*

$$\mathcal{D}^\perp = (P^{-1})^\top \mathcal{C}^\perp Q^\top.$$

Proof. Since $\mathcal{C}$ and $\mathcal{D}$ have the same dimension, so do $\mathcal{D}^\perp$ and $(P^{-1})^\top \mathcal{C}^\perp Q^\top$. Hence, it is enough to prove that one of these spaces is included in the other. Consider any $B \in \mathcal{D}$ and $A \in \mathcal{C}^\perp$. There is a matrix $C \in \mathcal{C}$ such that $B = PCQ^{-1}$. We have:

$$\begin{aligned}\mathrm{Tr}(B^\top (P^{-1})^\top A Q^\top) &= \mathrm{Tr}((Q^{-1})^\top C^\top P^\top (P^{-1})^\top A Q^\top) \\ &= \mathrm{Tr}((Q^{-1})^\top C^\top A Q^\top) \\ &= \mathrm{Tr}(C^\top A) = 0. && \text{(since } A \in \mathcal{C}^\perp\text{)}\end{aligned}$$

Hence, $\mathcal{D} \subseteq (P^{-1})^\top \mathcal{C}^\perp Q^\top$. □

Lemma 3. *Let $\mathcal{C}, \mathcal{D} \subset \mathbb{F}_q^{m\times m}$ be two $\mathbb{F}_q$-vector spaces. Let $P \in \mathrm{GL}_m(\mathbb{F}_q)$ be a matrix such that $\mathcal{D} = P\mathcal{C}P^{-1}$. Then*

$$h(\mathcal{D}) = P h(\mathcal{C}) P^{-1}.$$

Proof. Let $C \in h(\mathcal{C})$, and set $D = PCP^{-1} \in \mathcal{D}$. Let us show that $D \in h(\mathcal{D})$. Let $B \in \mathcal{D}$. There exists $A \in \mathcal{C}$ such that $B = PAP^{-1}$. We have

$$\begin{aligned}\mathrm{Tr}(BD) &= \mathrm{Tr}(PAP^{-1}PCP^{-1}) \\ &= \mathrm{Tr}(AC) \\ &= 0. && \text{(since } C \in h(\mathcal{C})\text{)}\end{aligned}$$

The other inclusion is proved in the same way. □

The following proposition, which says that roughly $1/q$ of all codes have a one-dimensional hull, is a consequence of results presented in [25]. It is explained in Appendix B, and will be crucial in the complexity analysis.

In the sequel, we denote by $\ker(\mathrm{Tr})$ the subspace of $\mathbb{F}_q^{m\times m}$ of matrices whose trace is zero.

Proposition 1. *The proportion of $m \times m$ matrix codes contained in* $\ker(\mathrm{Tr})$ *and whose hull has dimension 1 is asymptotically equal to*

$$\frac{1}{q}\left(1 + \mathcal{O}\left(\frac{m^2}{q^{(m^2-1)/2}}\right)\right).$$

2.2 Summary of the algorithm

We are given two k-dimensional subspaces $\mathcal{C}, \mathcal{D}$ of $\mathbb{F}_q^{m\times n}$. Our aim is to find two matrices $P \in \mathrm{GL}_m(\mathbb{F}_q)$ and $Q \in \mathrm{GL}_n(\mathbb{F}_q)$ verifying $\mathcal{D} = P\mathcal{C}Q^{-1}$. If we have found a suitable matrix P, computing Q can be done using linear algebra (see Section 5). The strategy for finding P consists first in guessing a pair $(A, B) \in \mathcal{C}^\perp \times \mathcal{D}^\perp$ such that $B = (P^{-1})^\top A Q^\top$, that is, a pair A, B which match with respect to the equivalence $\mathcal{D}^\perp = (P^{-1})^\top \mathcal{C}^\perp Q^\top$ given by Lemma 2. With such a pair at hand one can reduce the equivalence problem to the conjugacy problem of the codes

$$\mathcal{C}_A \stackrel{\text{def}}{=} \mathcal{C}A^\top \qquad \text{and} \qquad \mathcal{D}_B \stackrel{\text{def}}{=} \mathcal{D}B^\top. \tag{1}$$

Indeed, if $B = PAQ^{-1}$ we prove in Lemma 4 further that $\mathcal{D}_B = P\mathcal{C}_A P^{-1}$. Solving a matrix code conjugacy problem in this context is generally as hard as solving MCE [14], but it is easy in a particular case: when the hull of both codes has dimension 1, the generators of these two hulls are still conjugate. The main two steps in order to find P are the following.

1. From $\mathcal{C}, \mathcal{D}$, construct two conjugate codes $\mathcal{C}_A, \mathcal{D}_B$ with one-dimensional.
2. Compute a matrix R that conjugates these hulls and deduce a matrix P such that that $\mathcal{D}_B = P\mathcal{C}_A P^{-1}$.

First step. We begin by finding two matrices $A \in \mathcal{C}^\perp$ and $B \in \mathcal{D}^\perp$ such that the codes $\mathcal{C}_A$ and $\mathcal{D}_B$ of (1) have conjugate hulls. For any A, one may find at least one such B, which is $(P^{-1})^\top A Q^\top$.

In order to determine these matrices A and B, we construct a dictionary whose keys are (normalized and suitably chosen) polynomials $\chi \in \mathbb{F}_q[t]$ of degree m. The values corresponding to a key χ are the pairs $(A, U) \in \mathbb{F}_q^{m\times n} \times \mathbb{F}_q^{m\times m}$ such that the hull $h(\mathcal{C}_A)$ is one-dimensional and generated by the matrix U with characteristic polynomial χ. We construct this dictionary with roughly $q^{(m-3)/2}$ entries. Then, we apply the same process to $\mathcal{D}$ and look for collisions; the length of the lists ensures that we will find a pair (A, B). This step is explained in detail in Section 3.

Second step. Once we have a pair of matrices $(A, B) \in \mathcal{C}^\perp \times \mathcal{D}^\perp$ such that $h(\mathcal{C}_A)$ and $h(\mathcal{C}_B)$ are one-dimensional and generated by conjugate matrices U and V, we may easily compute a matrix $R \in \mathrm{GL}_m(\mathbb{F}_q)$ such that $V = RUR^{-1}$. We also impose in the collision search that the characteristic polynomial χ of U, V is squarefree so that U, V are both diagonalizable in an extension of $\mathbb{F}_q$. This allows us to reduce to the case of two codes whose one-dimensional hulls are spanned by the same diagonal matrix. In this context, we will observe that the matrix P we are looking for, *i.e.* the matrix P such that $\mathcal{D}_B = P\mathcal{C}_A P^{-1}$, is the product of R by some invertible matrix which can be expressed as $f(U)$ for some polynomial $f \in \mathbb{F}_q[t]$ of degree less than m. This polynomial f will be computed by looking at the action of P on a suitable subspace of $\mathcal{C}_A$ and $\mathcal{D}_B$. This step is explained in Section 4.

2.3 A comment on matrix code equivalence *v.s.* matrix code conjugacy

A remark that arises from our work, is that the equivalence problem seems to become much easier when reducing from general matrix code equivalence (*i.e.* arbitrary P, Q) to matrix code conjugacy (*i.e.* $m = n$ and $P = Q$). It is interesting to observe that from a complexity theory point of view the two problems are polynomially equivalent [14, Thm. A]. Still, the use of the hull gives a heuristic polynomial-time algorithm that solves a proportion $\mathcal{O}(1/q)$ of instances of the conjugacy problem (the $1/q$ coming from the fact that a random matrix code has a one-dimensional hull with probability $\mathcal{O}(1/q)$).

This phenomenon could be compared with what happens in classical coding theory, where two problems arise : the *permutation equivalence problem* (finding a permutation matrix sending a code to another) and the *monomial equivalence problem* (finding a monomial matrix, *i.e.* the product of a permutation matrix and a nonsingular diagonal matrix sending one code to another). When the ground field cardinality q is polynomial in the code length, the two problems are known to be polynomially equivalent [27] but Sendrier's *Support Splitting algorithm* is on average efficient on the former while being completely inefficient on the latter.

2.4 Complexity and impact

The complexity of the algorithm is dominated by the collision search in the first step. Given a uniformly random code $\mathcal{C}$ and a uniformly random full-rank matrix $A \in \mathcal{C}^\perp$, the codes $\mathcal{C}_A$ are uniformly distributed among the matrix codes in $\ker(\mathrm{Tr}) \subset \mathbb{F}_q^{m\times m}$. Among these, roughly $1/q$ have a one-dimensional hull. By an argument similar to the birthday paradox, computing two lists of length roughly $q^{(m-3)/2}$ is enough to find some collisions. This requires picking matrices A, and for each of these, computing the hull of $\mathcal{C}_A$. The total time complexity is

$$\mathcal{O}\left(\max(k,k^\perp)\ (nm^{\omega-1} + \max(k,k^\perp)m^2)\ q^{(m-1)/2}\ \ \max(1, q^{\min(k,k^\perp)-m+1})\right)$$

operations in $\mathbb{F}_q$, where $k^\perp \stackrel{\text{def}}{=} mn - k$ is the dimension of the dual code $\mathcal{C}^\perp$ and ω is the exponent of matrix multiplication. In the cubic case $m = n = k$, this complexity can be reduced to

$$\mathcal{O}\left(n^4 q^{(n+1)/2}\right).$$

The space complexity is

$$\mathcal{O}\left(m(m+n+1)q^{(m-3)/2}\right)$$

elements of $\mathbb{F}_q$, which is essentially the size of the computed dictionaries.

3 Reducing to probably conjugate spaces

We look at k-dimensional matrix codes inside $\mathbb{F}_q^{m\times n}$. Given a code of dimension k, we will denote by $k^\perp = mn - k$ the dimension of its dual.

3.1 Structure of the reduction

Lemma 2 shows that the instances $(\mathcal{C}, \mathcal{D})$ and $(\mathcal{C}^\perp, \mathcal{D}^\perp)$ of MCE are equivalent. In particular, for complexity reasons, we may switch $(\mathcal{C}, \mathcal{D})$ for $(\mathcal{C}^\perp, \mathcal{D}^\perp)$: we will systematically choose the instance with the highest dimension. Indeed, since collision search is performed on the dual codes, we fit in the situation where the codes have the smallest possible duals.

Lemma 2 also shows that given $A \in \mathcal{C}^\perp$, the matrix $B = (P^{-1})^\top A Q^\top$ belongs to $\mathcal{D}^\perp$. Thee key algorithm lies in the following lemma.

Lemma 4. *Let $(A, B) \in \mathcal{C}^\perp \times \mathcal{D}^\perp$ such that $B = (P^{-1})^\top A Q^\top$. Then, the codes*

$$\mathcal{C}_A \stackrel{def}{=} \mathcal{C} A^\top \quad \textit{and} \quad \mathcal{D}_B \stackrel{def}{=} \mathcal{D} B^\top.$$

satisfy

$$\mathcal{D}_B = P \mathcal{C}_A P^{-1}.$$

Proof. This is a straightforward calculation. □

The aim of the first step of our algorithm is to find pairs (A, B) such that $\mathcal{C}_A$ and $\mathcal{D}_B$ are two conjugate k-dimensional codes, in order to find P. Given any $\mathcal{C}_A$, $\mathcal{D}_B$, finding a matrix P such that $\mathcal{D}_B = P\mathcal{C}_A P^{-1}$ is complicated: this is the code conjugacy problem (see for instance [14]). However, it is much easier if one knows a distinguished pair of conjugate elements $U \in \mathcal{C}_A$, $V \in \mathcal{D}_B$. In order to find such a pair (U, V), we need to find conjugate one-dimensional subspaces in both $\mathcal{C}_A$ and $\mathcal{D}_B$. We can do this when the hulls of both $\mathcal{C}_A$ and $\mathcal{D}_B$ are one-dimensional, since, as shown in Lemma 3, the hulls of two conjugate matrix codes are conjugate.

Remark 6. By construction the code $\mathcal{C}_A \subset \ker(\mathrm{Tr})$. Moreover, in order to have one-dimensional hull, we need the former inclusion to be a strict. Since we require $\mathcal{C}_A$ to have the same dimension as $\mathcal{C}$, this means that our algorithm in this form only works for

$$k < m^2 - 1.$$

Allowing ourselves to switch $\mathcal{C}$ and $\mathcal{C}^\perp$ at the beginning, it also applies to the case where

$$k^\perp < m^2 - 1.$$

Given an instance $(\mathcal{C}, \mathcal{D})$ of $\mathsf{MCE}_{m,n,k}$, our goal is to find matrices $P \in \mathrm{GL}_m(\mathbb{F}_q)$ and $Q \in \mathrm{GL}_n(\mathbb{F}_q)$ such that $\mathcal{D} = P\mathcal{C}Q^{-1}$. The method below allows us to reduce the problem to the case where $m = n$, $P = Q$, and $\mathcal{C}$ and $\mathcal{D}$ have non trivial conjugate hulls. The reduction consists in the following steps.

1. Construct a dictionary $\{\chi : (A, U)\}$, where $\mathcal{C}_A = \mathcal{C}A^\top$ has a one–dimensional hull, $U \in \mathbb{F}_q^{m\times m} \setminus \{0\}$ generates this hull and $\chi \in \mathbb{F}_q[t]_{\leqslant m}$ is the characteristic polynomial of U, which we require to be squarefree. This is done in a very straightforward way: pick A at random, compute $h(\mathcal{C}_A)$ and if it is one-dimensional and generated by a matrix U, compute its characteristic polynomial χ_U and add the entry $(\chi\colon (A, U))$ to the dictionary. To make the second step easier, we only keep A when χ is separable. The precise procedure is explained in Algorithm 2. In order to find collisions more easily, we normalize the characteristic polynomials as explained in Appendix A: this reduces the number of possible characteristic polynomials to approximately q^{m-3} (see Lemma 14).

2. Pick random matrices $B \in \mathcal{D}^\perp$, and if the hull of $\mathcal{D}B^\top$ is one-dimensional, check if the characteristic polynomial of one of its generators is a key in the dictionary. The aforementioned conditions on the characteristic polynomials directly imply that the generators U, V of $h(\mathcal{C}_A), h(\mathcal{D}_B)$ having the same characteristic polynomial χ are conjugate: for each collision, we immediately compute $R \in \mathrm{GL}_m(\mathbb{F}_q)$ such that $V = RUR^{-1}$. The collision-finding procedure is described in Algorithm 3.

This yields a list of tuples (A, B, U, V, R) such that the codes $\mathcal{C}_A$ and $\mathcal{D}_B$ have one-dimensional hulls respectively generated by matrices U, V such that $V = RUR^{-1}$. To these tuples, we then apply the algorithm of Section 4 which allows to find a suitable matrix P such that $\mathcal{D}_B = P\mathcal{C}_A P^{-1}$. In order to compute Q, we now need to solve $\mathcal{D} = (P\mathcal{C}) \cdot Q$, where $P\mathcal{C}$ is known. This is an easy problem, which is solved by linear algebra as explained in Section 5.

Algorithm 1: ComputeCharpoly

Data: k-dimensional code $\mathcal{C} \subset \ker(\mathrm{Tr}) \subset \mathbb{F}_q^{m\times m}$ such that $\dim h(\mathcal{C}) = 1$
Result: Pair (χ, U) where $U \in \mathbb{F}_q^{m\times m}$ generates $h(\mathcal{C})$, and $\chi \in \mathbb{F}_q^{m-2} - \{0\}$ represents U's characteristic polynomial

Compute a generator U of $h(\mathcal{C})$
Compute char. polynomial $a_0 + a_1 t + \cdots + a_{m-3} t^{m-3} + t^m$ of U
Set $\chi = (a_{m-3}, a_{m-4}, \ldots, a_0) \in \mathbb{F}_q^{m-2} - \{0\}$
$\lambda =$ Normalize(U, χ) using Algorithm 6
return $(\lambda \diamond \chi, \lambda U)$ (where $\diamond$ is defined in $(\star\star\star)$ below)

Algorithm 1 computes the normalized generator of the hull of a code with one-dimensional hull.

Algorithm 2: CONSTRUCTDICT

Data: k-dimensional code $\mathcal{C} \subset \mathbb{F}_q^{m\times n}$, integer L
Result: Dictionary $\{\chi : (A, U)\}$ with L keys
where $U \in \mathbb{F}_q^{m\times m}$ generates $h(\mathcal{C}_A)$ and has characteristic polynomial χ

```
Dict = {}
Compute a basis of C^⊥
while length(Dict) < L do
    Pick a random A ∈ C^⊥
    if rk(A) = m and dim(C_A) = k then
        if dim h(C_A) = 1 then
            (χ, U) = ComputeCharpoly(C_A)
            if gcd(χ(t), χ'(t)) = 1 then
                Add entry (χ : (A, U)) to Dict
return Dict
```

Algorithm 2 constructs a dictionary whose keys are separable polynomials, and whose values are pairs of matrices (A, U) such that $h(\mathcal{C}_A) = \mathbb{F}_q U$.

Algorithm 3: FINDING COLLISIONS

Data: k-dimensional code $\mathcal{D} \subset \mathbb{F}_q^{m\times n}$, integer L, dictionary Dict
Result: List of tuples $(A, B, U, V, R) \in (\mathbb{F}_q^{m\times n})^2 \times (\mathbb{F}_q^{m\times m})^2 \times \mathrm{GL}_m(\mathbb{F}_q)$
where (A, U) is an item in Dict and $V = RUR^{-1}$ generates $h(\mathcal{D}_B)$

```
CollisionList = [ ]
Compute basis of C^⊥
while length(CollisionList) < L do
    Pick a random B ∈ D^⊥
    if rk(B) = m and dim(D_B) = k then
        Compute a basis of h(D_B)
        if dim h(D_B) = 1 then
            (χ, V) = ComputeCharpoly(D_B)
            if χ is a key of Dict then
                for (A, U) ∈ Dict[χ] do
                    Compute a matrix R ∈ GL_m(F_q) such that V = RUR^{-1}
                    Add (A, B, U, V, R) to CollisionList
return CollisionList
```

Algorithm 3 returns list of tuples (A, U, B, V, R) where $h(\mathcal{C}_A) = \mathbb{F}_q U$, $h(\mathcal{C}_B) = \mathbb{F}_q V$ and $V = RUR^{-1}$.

Remark 7. As presented, Algorithm 3 computes a whole list of collisions and then applies to each one the algorithms described in Section 4, until the code equivalence is solved. In practice, one can actually apply the algorithms of Section 4 on the fly each time a collision is found until we find a collision for which the algorithm of Section 4 yield the searched code equivalence.

3.2 Distribution of the computed matrix spaces and polynomials

In this section, we discuss the distribution of the matrix spaces and characteristic polynomials obtained using the algorithms above. Given a vector space V and an integer d, we denote by $\mathrm{Gr}_d(V)$ (resp. $\mathrm{Gr}_{\leqslant d}(V)$) the set of all d-dimensional (resp. at most d-dimensional) linear subspaces of V. We prove that given a uniformly random $\mathcal{C} \in \mathrm{Gr}_k(\mathbb{F}_q^{m\times n})$ and $A \in \mathcal{C}^\perp$ such that $\mathcal{C}_A$ has one-dimensional hull (and some mild additional conditions), the distribution of the characteristic polynomials of a generator of these hulls is asymptotically uniform with respect to q.

Given a matrix $A \in \mathbb{F}_q^{m\times n}$, we define the map

$$\begin{aligned}\phi_A \colon \mathbb{F}_q^{m\times n} &\longrightarrow \mathbb{F}_q^{m\times m}\\ M &\longmapsto MA^\top.\end{aligned}$$

For a k-dimensional code $\mathcal{C} \subset \mathbb{F}_q^{m\times n}$ and a matrix $A \in \mathbb{F}_q^{m\times n}$, we consider in our reduction the code $\phi_A(\mathcal{C}) = \mathcal{C}_A \subset \mathbb{F}_q^{m\times m}$. This amounts to considering the map

$$\begin{aligned}\Phi\colon\ \mathrm{Gr}_k(\mathbb{F}_q^{m\times n}) \times \mathbb{F}_q^{m\times n} &\longrightarrow \mathrm{Gr}_{\leqslant k}(\mathbb{F}_q^{m\times m})\\ (\mathcal{C}, A) &\longmapsto \phi_A(\mathcal{C}) = \mathcal{C}_A.\end{aligned}$$

We will choose A to have full rank m (recall that $m \leqslant n$). This entails that ϕ_A is surjective. The preimages of a k-dimensional code $\mathcal{D} \subset \mathbb{F}_q^{m\times m}$ under Φ are exactly the pairs $(\mathcal{C}, A)$ such that $\mathcal{C} \subset \phi_A^{-1}(\mathcal{D})$ and $\mathcal{C} \cap \ker(\phi_A) = 0$. We now set

$$X \stackrel{\text{def}}{=} \{(\mathcal{C}, A) \in \mathrm{Gr}_k(\mathbb{F}_q^{m\times n}) \times \mathbb{F}_q^{m\times n} \mid A \in \mathcal{C}^\perp,\ \mathrm{rk}(A) = m,\ \mathcal{C} \cap \ker(\phi_A) = 0\}.$$

Lemma 5. *The restricted map*

$$f_1 = \Phi_{|X} \colon X \to \mathrm{Gr}_k(\ker(\mathrm{Tr})), \tag{$\star$}$$

where $\mathrm{Gr}_k(\ker(\mathrm{Tr}))$ *denotes the set of k–dimensional spaces of $m \times m$ matrices whose trace is zero, is surjective and equidistributed (i.e. each element of its image has the same number of preimages).*

Proof. Given $\mathcal{D} \in \mathrm{Gr}_k(\mathbb{F}_q^{m\times m})$ whose elements have trace zero, any element $(\mathcal{C}, A) \in \Phi_{|X}^{-1}(\mathcal{D})$ satisfies $A \in \mathcal{C}^\perp$. Given a rank m matrix $A \in \mathbb{F}_q^{m\times n}$, the codes $\mathcal{C}$ such that $\mathcal{C}A^\top = \mathcal{D}$ and $(\mathcal{C}, A) \in X$ are exactly the complementary subspaces of $\ker(\phi_A)$ in $\phi_A^{-1}(\mathcal{D})$. The number of elements in $\Phi_{|X}^{-1}(\mathcal{D})$ is the number of rank m matrices in $\mathbb{F}_q^{m\times n}$ multiplied by the number of complementary subspaces of an $m(n-m)$-dimensional subspace in an $(k + m(n-m))$-dimensional $\mathbb{F}_q$-vector space. The latter number is nonzero and does not depend on a particular choice of $\mathcal{D}$. Hence, the map f_1 is surjective and equidistributed. □

Lemma 6. *In a code $\mathcal{C}_A \subset \ker(\mathrm{Tr})$ any element in the hull of $\mathcal{C}_A$ satisfies $\mathrm{Tr}(U^2) = 0$.*

Proof. This is a direct consequence of the definition of the hull (Definition 5.

Lemma 7. *If q is not a power of 2, the map*

$$f_2 \colon \{\mathcal{C} \in \mathrm{Gr}_k(\ker(\mathrm{Tr})) \mid \dim h(\mathcal{C}) = 1\} \to \{U \in \mathbb{F}_q^{m\times m} \mid \mathrm{Tr}(U) = \mathrm{Tr}(U^2) = 0\}/\mathbb{F}_q^\times \qquad (\star\star)$$

which sends $\mathcal{C}$ to a generator of $h(\mathcal{C})$ (modulo the action of $\mathbb{F}_q^\times$) is equidistributed. In particular, as soon as the set on the left is nonempty, f_2 is surjective.

Proof. We work in $\ker(\mathrm{Tr})$ with the non-degenerate bilinear form $(X, Y) \mapsto \mathrm{Tr}(XY)$. Consider any two matrices U_1, U_2 such that $\mathrm{Tr}(U_i) = \mathrm{Tr}(U_i^2) = 0$. To prove equidistribution, it is enough to construct a bijection between their preimages under this map. The map $\mathbb{F}_q U_1 \to \mathbb{F}_q U_2$ which sends U_1 to U_2 is an isometry with respect to the aforementioned bilinear form. Since $\mathrm{char}(\mathbb{F}_q) \neq 2$, Witt's extension theorem [15, Thm. 5.2] ensures that this map extends to an isometry g of $\ker(\mathrm{Tr})$. Then, the map

$$\{\mathcal{C} \subset \ker(\mathrm{Tr}) \mid h(\mathcal{C}) = \mathbb{F}_q U_1\} \to \{\mathcal{C} \subset \ker(\mathrm{Tr}) \mid h(\mathcal{C}) = \mathbb{F}_q U_2\}$$

which sends $\mathcal{C}$ to $g(\mathcal{C})$ is a bijection. □

Remark 8. Recall that Proposition 1 states that asymptotically, $1/q$ of all matrix codes in $\mathrm{Gr}_k(\ker \mathrm{Tr})$ have a one-dimensional hull. Hence, the number of these codes is equivalent to $q^{k(m^2-1-k)-1}$. Therefore, for big enough q, such codes exist, and the map f_2 is always surjective.

Lemma 8. *Let $\chi \in \mathbb{F}_q[t]$ be a separable polynomial of degree m such that $\chi(0) \neq 0$. The number of matrices U with characteristic polynomial χ is asymptotically (when $q \to \infty$) equivalent to q^{m^2-m}.*

Proof. Let U be a matrix with characteristic polynomial χ. Since χ is separable, it is also the minimal polynomial of U, and the matrices with characteristic polynomial χ are conjugates of U. There are as many conjugates of U as elements in the quotient

$$\mathrm{GL}_m(\mathbb{F}_q)/\{P \in \mathrm{GL}_m(\mathbb{F}_q) \mid PUP^{-1} = U\}.$$

Since U has a separable characteristic polynomial, any matrix which commutes with U is a polynomial in U [16, Cor. IV.E.8]. We are looking for the cardinality of $\mathbb{F}_q[U] \cap \mathrm{GL}_m(\mathbb{F}_q)$. A classical consequence of Cayley Hamilton theorem entails that $\mathbb{F}_q[U] \cap \mathrm{GL}_m(\mathbb{F}_q)$ is nothing but the group $\mathbb{F}_q[U]^\times$ of invertible elements of the ring $\mathbb{F}_q[U]$. Hence, the polynomials $f \in \mathbb{F}_q[t]/(\chi)$ such that $f(U)$ is not invertible are those that are divisible by an irreducible factor of χ. Their number is maximal when U is diagonalizable over $\mathbb{F}_q$, in which case there are less than mq^{m-1} such polynomials. Hence, $\mathbb{F}_q[U] \cap \mathrm{GL}_m(\mathbb{F}_q)$ has at least $q^m - mq^{m-1} = q^m(1 - m/q)$ elements; since it always has less than q^m elements, its cardinality is equivalent to q^m, and that of $\mathrm{GL}_m(\mathbb{F}_q)/(\mathbb{F}_q[U] \cap \mathrm{GL}_m(\mathbb{F}_q))$ is equivalent to q^{m^2-m}. □

Consider the set of matrices $U \in \mathrm{GL}_m(\mathbb{F}_q)$ such that $\mathrm{Tr}(U) = \mathrm{Tr}(U^2) = 0$, up to scalar multiplication. The characteristic polynomial $\chi_U \in \mathbb{F}_q[t]$ of such a matrix U is of the form $t^m + a_{m-3}t^{m-3} + \cdots + a_1 t + a_0$, with $a_0 \neq 0$ since U is invertible. For any $\lambda \in \mathbb{F}_q^\times$, the characteristic polynomial of λU is

$$\chi_{\lambda U} = t^m + \lambda^3 a_{m-3} + \cdots + \lambda^{m-1} a_1 + \lambda^m a_0.$$

Hence, there is a map

$$\begin{aligned} f_3 \colon \{U \in \mathrm{GL}_m(\mathbb{F}_q) \mid \mathrm{Tr}(U) = \mathrm{Tr}(U^2) = 0\}/\mathbb{F}_q^\times &\longrightarrow \mathbb{F}_q^{m-2}/\mathbb{F}_q^\times \\ U &\longmapsto (a_{m-3}, \ldots, a_0) \end{aligned} \qquad (\star\star\star)$$

where $\mathbb{F}_q^\times$ acts on $\mathbb{F}_q^{m-2}$ via $\lambda \diamond (a_{m-3}, \ldots, a_0) = (\lambda^3 a_{m-3}, \ldots, \lambda^m a_0)$. Its image is $(\mathbb{F}_q^{m-2} - (\mathbb{F}_q^{m-1} \times \{0\}))/\mathbb{F}_q^\times$. Lemma 8 asserts that any element of the form $(a_{m-3}, \ldots, u_0)$ corresponding to a separable polynomial has $\sim q^{m^2-m}$ preimages under f_3.

Denote by

$$\mathrm{Sep}_{q,m} \subset (\mathbb{F}_q^{m-2} - (\mathbb{F}_q^{m-1} \times \{0\}))/\mathbb{F}_q^\times$$

the set of classes of separable characteristic polynomials with nonzero constant coefficient.

Remark 9. We give more details about this construction in Appendix A. In particular, we show in Lemma 14 that the set $\mathrm{Sep}_{q,m}$ has $\sim q^{m-3}$ elements. In Algorithm 1, we use a unique representative of each class of characteristic polynomials. The way of computing such a normalized representative is also explained in Appendix A and presented in Algorithm 6.

Proposition 2. *Suppose q is not a power of 2. Denote by*

$$f_q = f_3 \circ f_2 \circ f_1 \colon X \to \mathbb{F}_q^{m-2}/\mathbb{F}_q^\times$$

the map which sends $(\mathcal{C}, A)$ to the equivalence class of the tuple of coefficients of the characteristic polynomial of a generator of the hull of $\mathcal{C}_A$. The maps f_1, f_2, f_3 are defined in $(\star), (\star\star), (\star\star\star)$. The map

$$f_{q|f_q^{-1}(\mathrm{Sep}_{q,m})} \colon f_q^{-1}(\mathrm{Sep}_{q,m}) \to \mathrm{Sep}_{q,m}$$

is asymptotically equidistributed, i.e.

$$\min_{\chi \in \mathrm{Sep}_{q,m}} |f_q^{-1}(\chi)| \underset{q\to\infty}{\sim} \max_{\chi \in \mathrm{Sep}_{q,m}} |f_q^{-1}(\chi)|.$$

Proof. The map f_3 is asymptotically equidistributed on $f_3^{-1}(\mathrm{Sep})$ by Lemma 8. The map f_2 is equidistributed by Lemma 7, and f_1 is equidistributed by Lemma 5. $\square$

Remark 10. The elements produced by Algorithm 2 are characteristic polynomials obtained by picking uniformly random elements of $f_q^{-1}(\mathrm{Sep}_{q,m})$ and computing a normalized representative of their image under the map f. Hence, Proposition 2 shows that given uniformly random inputs $\mathcal{C} \in \mathrm{Gr}_k(\mathbb{F}_q^{m\times n})$, the distribution of normalized characteristic polynomials $\chi \in \mathrm{Sep}_{q,m}$ produced by Algorithm 2 is asymptotically uniform.

Remark 11. The result above shows that the distribution of the computed characteristic polynomials is asymptotically uniform for random $\mathcal{C}$ and A. But in practice, a fixed code $\mathcal{C}$ is given to us. In that case, we have not said anything about the distribution of the codes $\mathcal{C}A^\top$ yet. In the special case $m = n$, the map

$$\psi_{\mathcal{C}} \colon \mathcal{C}^\perp \cap \mathrm{GL}_m(\mathbb{F}_q) \to \mathrm{Gr}_k(\mathbb{F}_q^{m\times m})$$

sending A to $\mathcal{C}A^\top$ is equidistributed. Indeed, given two codes $\mathcal{D}_1 = \mathcal{C}A_1^\top \in \mathrm{Gr}_k(\mathbb{F}_q^{m\times m})$ and $\mathcal{D}_2 = \mathcal{C}A_2^\top$, the map

$$\begin{aligned} \psi_{\mathcal{C}}^{-1}(\mathcal{D}_1) &\longrightarrow \psi_{\mathcal{C}}^{-1}(\mathcal{D}_2) \\ B_1 &\longmapsto A_2 A_1^{-1} B_1 \end{aligned}$$

is a bijection.

3.3 Complexity analysis

In this section, the symbol $\sim$ always denotes asymptotic equivalence, and the notation $o(\cdot)$ denotes asymptotic domination, with respect to the parameter q. We make the following assumption, justified by Remarks 10 and 11.

Assumption 1. *Given a code $\mathcal{C}$, distinct full-rank matrices A yield distinct codes $\mathcal{C}_A = \mathcal{C}A^\top$ and the characteristic polynomials χ are uniformly distributed among the codes $\mathcal{C}_A$ with one-dimensional hull. For this, we require $k \leqslant m^2 - 2$: otherwise, the codes $\mathcal{C}A^\top$ would be the full* $\ker(\mathrm{Tr})$ *as soon as A has full rank, and could not have a one-dimensional hull.*

We are going to answer the following questions:

1. How many collisions do we need to find?
2. How many matrices A do we need to sample to find enough characteristic polynomials?
3. How many operations are needed to compute the dictionary?
4. What is the total complexity of running Algorithms 2 and 3 with the parameters answering the previous questions?

How many collisions do we need to find? For any $A \in \mathcal{C}^\perp$ and any $\lambda \in \mathbb{F}_q^\times$, $\mathcal{C}_A = \mathcal{C}_{\lambda A}$. Hence, the total number of codes $\mathcal{C}_A$, $A \in \mathcal{C}^\perp$ is less than

$$\frac{1}{q-1}(\#\mathcal{C}^\perp - 1) = \frac{q^{k^\perp} - 1}{q-1} \sim q^{k^\perp - 1}$$

where $k^{\perp} = \dim(\mathcal{C}^{\perp}) = mn - k$. The number of $\mathcal{C}_A$ with one-dimensional hull is therefore equivalent to $q^{k^{\perp}-2}$ by Proposition 1. The total number of possible classes of separable characteristic polynomials is $\sim q^{m-3}$ (see Lemma 14). Thus, given a code $\mathcal{C}_A$, there are $\sim q^{k^{\perp}-m+1}$ codes $\mathcal{D}_B$ that yield the same characteristic polynomial. One of these is the matrix $(P^{-1})^{\top} A Q^{\top}$ we are looking for. This means that we have to list on average $\sim \max(1, q^{k^{\perp}-m+1})$ collisions in order to find the right one.

How many matrices A do we need to sample in order to find enough characteristic polynomials?

Lemma 9. *Let $r = o(q^{m-3})$ be an integer. The average number of matrices to sample in Algorithm 2 in order to get r distinct characteristic polynomials is $\sim qr$.*

Proof. This is a variant of the coupon collector's problem. Denote by

$$N_{\chi} \sim q^{m-3}$$

the total number of possible characteristic polynomials with the shape $X^m + a_{m-3}X^{m-3} + \cdots + a_0$, by $M \leqslant (q^{k^{\perp}} - 1)/(q-1)$ the number of elements in $(\mathcal{C}^{\perp} - \{0\})/\mathbb{F}_q^{\times}$ with full rank and by S_r the number of matrices A we have to sample in order to get r different characteristic polynomials of matrices spanning one-dimensional hulls of codes $\mathcal{C}_A$. Denote by s_j the number of matrices to sample after having a list of $j-1$ distinct polynomials in order to get the j-th one. We seek to compute the expected value

$$\mathbb{E}(S_r) = \mathbb{E}(s_1) + \cdots + \mathbb{E}(s_r).$$

The random variable s_j follows a geometric distribution: it is the first success of a Bernoulli variable. The parameter p_j of this variable is computed as follows: it is the proportion, among all the elements of $(\mathcal{C}^{\perp} - \{0\})/\mathbb{F}_q^{\times}$, of those (equivalence classes of) matrices A yielding a code $\mathcal{C}_A$ with one-dimensional hull and characteristic polynomial that is not among the j polynomials already in the list. The number of full–rank matrices A that yield a code $\mathcal{C}_A$ with one-dimensional hull is

$$M\left(\frac{1}{q} + \mathcal{O}\left(\frac{m^2}{q^{(m^2+1)/2}}\right)\right).$$

Under Assumption 1, the number of matrices that yield a code $\mathcal{C}_A$ with a one-dimensional hull and one of the j characteristic polynomials already in the list is

$$j \cdot \frac{M\left(\frac{1}{q} + \mathcal{O}\left(\frac{m^2}{q^{(m^2+1)/2}}\right)\right)}{N_{\chi}}.$$

Hence

$$\begin{aligned}
p_j &= \frac{1}{M}\left[\frac{M}{q} + \mathcal{O}\left(\frac{m^2 M}{q^{(m^2+1)/2}}\right) - j\frac{M}{qN_\chi} + \frac{j}{N_\chi}\mathcal{O}\left(\frac{m^2 M}{q^{(m^2+1)/2}}\right)\right] \\
&= \frac{1}{q}\left(1 - \frac{j}{N_\chi}\right) + \mathcal{O}\left(\frac{m^2}{q^{(m^2+1)/2}}\right) \\
&\sim \frac{1}{q}\left(1 - \frac{j}{N_\chi}\right) \qquad\qquad \left(\text{since } \tfrac{1}{qN_\chi} \sim \tfrac{1}{q^{m-2}}\right) \\
&\sim \frac{N_\chi - j}{qN_\chi}.
\end{aligned}$$

The expected value of the geometric random variable s_j with parameter p_j is $1/p_j$. Hence, using the fact that $r = o(N_\chi)$,

$$\begin{aligned}
\mathbb{E}(S_r) &\sim qN_\chi\left(\frac{1}{N_\chi} + \cdots + \frac{1}{N_\chi - r + 1}\right) \\
&\sim qN_\chi \log\left(\frac{N_\chi}{N_\chi - r}\right) \\
&\sim -qN_\chi \log\left(1 - r/N_\chi\right) \\
&\sim qr.
\end{aligned}$$

□

Complexity of computing the dictionary

Lemma 10. *The average complexity of Algorithm 2 with input a k-dimensional code $\mathcal{C} \subset \mathbb{F}_q^{m\times n}$ and a desired list length $L = o(q^{m-3})$ is*

$$\mathcal{O}(qLk(nm^{\omega-1} + km^2)).$$

Proof. In order to get L distinct characteristic polynomials, Lemma 9 tells us that we need to sample $\approx qL$ matrices A. For each of these, we first need to compute a basis $(C_1, \ldots, C_k)$ of $\mathcal{C}A^\top$, which is given by k products of a matrix of size $m \times n$ by a matrix of size $n \times m$; this requires $\mathcal{O}(knm^{\omega-1})$ operations in $\mathbb{F}_q$. Then, we need to compute $\mathcal{C}_A \cap \mathcal{C}_A^\perp$, which is given by the kernel of the (symmetric) Gram matrix $(\mathrm{Tr}(C_iC_j))_{1\leqslant i,j\leqslant k}$. Computing the diagonal entries of a given product C_iC_j requires $\mathcal{O}(m^2)$ operations in $\mathbb{F}_q$. Hence, the Gram matrix is computed in $\mathcal{O}(k^2m^2)$ operations in $\mathbb{F}_q$. Computing its kernel takes $\mathcal{O}(k^\omega)$ operations in $\mathbb{F}_q$. When the hull has dimension 1, we then only need to compute the characteristic polynomial of the generator we have found, which is done in $\mathcal{O}(m^\omega)$ operations [21, Thm. 1.1], and to normalize it. This normalization can be precomputed for a proportion $(1-2/q)$ of all cases (see Remark 15). So sampling qL matrices to find enough hulls $h(\mathcal{C}_A)$ takes $\mathcal{O}(qL(knm^{\omega-1} + k^2m^2 + k^\omega + m^\omega))$ operations. The result follows from the fact that $k \leqslant m^2$ and $m \leqslant n$. □

Corollary 1. *The average complexity of Algorithm 3 with input a k-dimensional code $\mathcal{D} \subset \mathbb{F}_q^{m\times n}$, a dictionary of length $L \geqslant q$ and a desired number c of collisions is*

$$\mathcal{O}(cq^{m-2}k(nm^{\omega-1} + km^2)/L).$$

Proof. In order to get c collisions with keys of the dictionary, which are chosen uniformly within $\sim q^{m-3}$ elements, we need to construct a list of $\sim cq^{m-3}/L$ characteristic polynomials. The complexity of this was computed in the previous lemma. □

Total complexity of this reduction step Recall that we showed earlier (see beginning of Section 3.3) that we need to find on average $\sim \max(1, q^{k^\perp - m+1})$ collisions in order to find a suitable pair (A, B) such that $\mathcal{C}_A$ and $\mathcal{D}_B$ are actually conjugate.

In the following, we will allow ourselves to replace $\mathcal{C}$ with $\mathcal{C}^\perp$ if needed. Recall that we need $k < m^2 - 1$ for the algorithm to work (see Remark 6). Hence, if we want to switch $\mathcal{C}$ and $\mathcal{C}^\perp$, we need $k^\perp < m^2 - 1$, i.e. $k > m(n - m) + 1$.

Proposition 3. *Suppose that either $k^\perp \leqslant k < m^2-1$, or $k \leqslant k^\perp < m^2-1$. The reduction step, i.e. running Algorithm 2 to get a dictionary of length $q^{(m-3)/2}$ and Algorithm 3 to get $\max(1, q^{\min(k,k^\perp)-m+1})$ collisions takes an expected complexity of*

$$\mathcal{O}\left(\max(k, k^\perp)(nm^{\omega-1} + \max(k, k^\perp)m^2)q^{(m-1)/2} \max(1, q^{\min(k,k^\perp)-m+1})\right)$$

operations in $\mathbb{F}_q$, and a space complexity of

$$\mathcal{O}\left(m(n + m + 1)q^{(m-3)/2}\right)$$

elements of $\mathbb{F}_q$.

Proof. We may suppose $k^\perp \leqslant k$; if not, switch $\mathcal{C}$ and $\mathcal{C}^\perp$, $\mathcal{D}$ and $\mathcal{D}^\perp$. We need to find $\sim \max(1, q^{k^\perp-m+1})$ collisions among q^{m-3} elements. For this, we first compute a dictionary of length $q^{(m-3)/2}$ using Algorithm 2, then find $\max(1, q^{k^\perp-m+1})$ collisions using Algorithm 3. The result is a consequence of Lemma 10 and Corollary 1. The space complexity is simple to compute: it is dominated by the number of elements of $\mathbb{F}_q$ in the dictionary. There are $q^{(m-3)/2}$ entries in the dictionary, and each of them contains an $m \times n$ matrix, an $m \times m$ matrix, and coefficients of a polynomial of degree m. □

Remark 12. When $n = m$, this complexity can be further reduced by considering the codes

$$\mathcal{C}_A = \mathcal{C}A^{-1} \quad \text{for} \quad A \in \mathcal{C} \cap \mathrm{GL}_m(\mathbb{F}_q),$$

instead of $\mathcal{C}A^\top$ for $A \in \mathcal{C}^\perp$. The remainder of the algorithm is unchanged. The difference is that the algorithm now only requires

$$\mathcal{O}\left(\min(k, k^\perp)(nm^{\omega-1} + \min(k, k^\perp)m^2)q^{(m-1)/2} \max(1, q^{\min(k,k^\perp)-m+1})\right)$$

operations in $\mathbb{F}_q$.

Remark 13. When $m = n = k$, this is even simpler, as $\min(k, k^\perp) = k$. In that case, our algorithm requires

$$\mathcal{O}\left(n^4 q^{(n+1)/2}\right)$$

operations in $\mathbb{F}_q$.

4 Finding the right matrix

I order to shorten the notations, we now denote by $\mathcal{C}, \mathcal{D}$ the codes $\mathcal{C}_A, \mathcal{D}_B$. We are in the following situation: we are given two codes $\mathcal{C}, \mathcal{D} \subset \mathbb{F}_q^{m\times m}$ with one-dimensional hulls generated respectively by matrices U, V and a matrix $R \in \mathrm{GL}_m(\mathbb{F}_q)$ such that $V = RUR^{-1}$. Our aim is to decide whether $\mathcal{C}_A$ and $\mathcal{D}_B$ are conjugate by some matrix $P \in \mathrm{GL}_m(\mathbb{F}_q)$, and if they are, find such a matrix.

Our strategy may be broken down into the following two steps:

1. Since their characteristic polynomial is separable, U and V are diagonalizable over an extension of $\mathbb{F}_q$: we may reduce to the case where they are diagonal.
2. We find the matrix P, which is R multiplied by some element of $\mathbb{F}_q[U]$, by considering its action on some subspaces of $\mathcal{C}$.

4.1 Reducing to diagonal matrices

The one-dimensional hulls of the codes $\mathcal{C}$ and $\mathcal{D}$ are respectively generated by conjugate diagonalizable matrices $U, V \in \mathrm{GL}_m(\mathbb{F}_q)$. We can compute a matrix $R \in \mathrm{GL}_m(\mathbb{F}_q)$ such that $V = RUR^{-1}$, hence $h(\mathcal{D}) = Rh(\mathcal{C})R^{-1}$. We are looking for a matrix $P \in \mathbb{F}_q^{m\times m}$ such that $\mathcal{D} = P\mathcal{C}P^{-1}$.

We know that the matrix $P \in \mathbb{F}_q^{m\times m}$ we are looking for satisfies $V = PUP^{-1}$. Therefore, there exists a matrix $T \in \mathbb{F}_q^{m\times m}$ which commutes with U such that $P = RT$. Since the characteristic polynomial of U is separable, we may write $T = f(U)$, where $f = \alpha_0 + \alpha_1 t + \cdots + \alpha_{m-1}t^{m-1} \in \mathbb{F}_q[t]$ (see [16, Cor. IV.E.8]). We know R and search for a polynomial f such that

$$\mathcal{D} = Rf(U)\mathcal{C}f(U)^{-1}R^{-1}. \qquad (\circ)$$

Under the assumption that U, V have a squarefree characteristic polynomial, there is a diagonal matrix Δ and a matrix $S \in \mathrm{GL}_m(\mathbb{F}_q)$ both possibly defined over an extension $\mathbb{F}_{q'}$ of $\mathbb{F}_q$ such that $V = S\Delta S^{-1}$. Such matrices S, Δ are easily computable. Therefore,

$$U = R^{-1}VR = R^{-1}S\Delta S^{-1}R$$

and $(\circ)$ is equivalent to

$$\mathcal{D} = Sf(\Delta)S^{-1}R\mathcal{C}R^{-1}Sf(\Delta)^{-1}S^{-1}$$

i.e.,

$$S^{-1}\mathcal{D}S = f(\Delta)\cdot S^{-1}R\mathcal{C}R^{-1}S\cdot f(\Delta)^{-1}.$$

We may compute bases of $\mathcal{D}' = S^{-1}\mathcal{D}S$ and $\mathcal{C}' = S^{-1}R\mathcal{C}R^{-1}S$. The problem at hand is now to compute $f \in \mathbb{F}_q[t]$ of degree at most $m-1$ such that, given codes $\mathcal{C}', \mathcal{D}' \subset \mathbb{F}_q^{m\times m}$ of dimension k and a diagonal matrix Δ,

$$\mathcal{D}' = f(\Delta)\mathcal{C}'f(\Delta)^{-1}.$$

The reduction is summed up in the algorithm below.

Algorithm 4: Reducing to diagonal matrices

Data: Codes $\mathcal{C}, \mathcal{D} \subset \mathbb{F}_q^{m\times m}$
Matrices $U, V \in \mathbb{F}_q^{m\times m}$ with separable characteristic polynomial s.t. $h(\mathcal{C}) = \mathbb{F}_q \cdot U$ and $h(\mathcal{D}) = \mathbb{F}_q \cdot V$
Matrix $R \in \mathrm{GL}_m(\mathbb{F}_q)$ such that $V = RUR^{-1}$
Result: Field extension $\mathbb{F}_{q'}$ of $\mathbb{F}_q$ and tuple $(\mathcal{C}', \mathcal{D}', \Delta)$ where:
$\Delta \in \mathbb{F}_{q'}^{m\times m}$ is diagonal and conjugate to U, V
$\exists f \in \mathbb{F}_{q'}[t]$ s.t. $\mathcal{D}' = f(\Delta)\mathcal{C}'f(\Delta)^{-1} \subset \mathbb{F}_{q'}^{m\times m}$

Compute field extension $\mathbb{F}_{q'}$ of $\mathbb{F}_q$ over which U is diagonalizable
Compute $S \in \mathrm{GL}_m(\mathbb{F}_{q'})$ and diagonal $\Delta \in \mathbb{F}_{q'}^{m\times m}$ s.t. $V = S\Delta S^{-1}$
Compute bases of $\mathcal{C}' = S^{-1}R\mathcal{C}R^{-1}S$ and $\mathcal{D}' = S^{-1}\mathcal{D}S$
return $(\mathcal{C}', \mathcal{D}', \Delta)$

4.2 Conjugating by the right matrix

Replacing $q, \mathcal{C}, \mathcal{D}$ with $q', \mathcal{C}', \mathcal{D}'$, we are now left with the following problem. We are given codes $\mathcal{C}, \mathcal{D} \subset \mathbb{F}_q^{m\times m}$ of dimension k and a diagonal matrix Δ, and need to find a polynomial $f \in \mathbb{F}_q[t]$ such that $\deg(f) < m$ and $\mathcal{D} = f(\Delta)\mathcal{C}f(\Delta)^{-1}$. Note that if a polynomial f verifies this, any scalar multiple of f does, so we may assume that f is monic. We may write

$$\Delta = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_m \end{pmatrix}$$

and since Δ is diagonal,

$$f(\Delta) = \begin{pmatrix} f(\lambda_1) & & & \\ & f(\lambda_2) & & \\ & & \ddots & \\ & & & f(\lambda_m) \end{pmatrix}.$$

Our strategy is the following:

- Find the coefficients of $f(\Delta)$.
- Knowing Δ, retrieve f using Lagrange interpolation.

We may easily find a set Λ of $k-1$ non-diagonal indexes $(i,j) \in \{1 \dots m\}^2$ such that the respective intersections $\mathcal{C}(\Lambda), \mathcal{D}(\Lambda)$ of $\mathcal{C}, \mathcal{D}$ with the subspace E_Λ of $\mathbb{F}_q^{m \times m}$ defined by the equations $\{x_{i,j} = 0\}_{(i,j) \in \Lambda}$ are one-dimensional.

Lemma 11. *The subspaces $\mathcal{C}(\Lambda), \mathcal{D}(\Lambda)$ satisfy*

$$f(\Delta)\mathcal{C}(\Lambda)f(\Delta)^{-1} = \mathcal{D}(\Lambda).$$

Proof. Since Δ is diagonal, so is $f(\Delta)$, and conjugating a matrix by $f(\Delta)$ does not change those of its entries which are equal to zero. Hence,

$$f(\Delta)E_\Lambda f(\Delta)^{-1} = E_\Lambda.$$

The result now follows from the equalities below.

$$\begin{aligned} f(\Delta)\mathcal{C}(\Lambda)f(\Delta)^{-1} &= f(\Delta)(\mathcal{C} \cap E_\Lambda)f(\Delta)^{-1} \\ &= (f(\Delta)\mathcal{C}f(\Delta)^{-1}) \cap (f(\Delta)E_\Lambda f(\Delta)^{-1}) \\ &= (f(\Delta)\mathcal{C}f(\Delta)^{-1}) \cap E_\Lambda \\ &= \mathcal{D}(\Lambda). \end{aligned}$$

□

We now pick a matrix $C = (c_{ij})_{i,j} \in \mathcal{C}(\Lambda)$, and a matrix $D = (d_{ij})_{i,j} \in \mathcal{D}(\Lambda)$. After multiplying by an element of $\mathbb{F}_q$, we may suppose that they have the same characteristic polynomial, and solve $D = f(\Delta)Cf(\Delta)^{-1}$. This means solving the system of $m(m-1)/2 - (k-1)$ equations

$$d_{ij} = f(\lambda_i)f(\lambda_j)^{-1}c_{ij} \qquad (1 \leqslant i < j \leqslant m, (i,j) \notin S)$$

which yields the m values $f(\lambda_1), \dots, f(\lambda_m)$ up to a scalar multiple. We then find a polynomial f corresponding to these values using Lagrange interpolation.

Remark 14. Note that in some rare cases, in particular if k is very small, the matrix C could have too many zeros for the system to determine f uniquely. In that case, picking another set S of coordinates does the trick.

Algorithm 5: Find the right polynomial

Data: Codes $\mathcal{C}, \mathcal{D} \subset \mathbb{F}_q^{m \times m}$
Diagonal $\Delta \in \mathbb{F}_q^{m \times m}$ such that $\exists f \in \mathbb{F}_q[t]\colon \mathcal{D} = f(\Delta)\mathcal{C}f(\Delta)^{-1}$
Result: Polynomial $f \in \mathbb{F}_q[t]$ such that $\mathcal{D} = f(\Delta)\mathcal{C}f(\Delta)^{-1}$

```
while true do
    Pick set Λ of k − 1 random non diagonal indexes (i, j) ∈ {1, . . . , m}^2
    Compute C(Λ) = C ∩ {x_ij = 0}_(i,j)∈Λ, D(Λ) = D ∩ {x_ij = 0}_(i,j)∈Λ
    if dim C(Λ) = dim D(Λ) = 1 then
        Pick C ∈ C(Λ), D ∈ D(Λ) with the same characteristic
          polynomial
        Solve system d_ij = u_i u_j^{-1} c_ij for (i, j) ∈ {1 . . . m}^2
        Compute polynomial f such that f(δ_ii) = u_i
        return f
```

4.3 Complexity analysis

Lemma 12. *Given conjugate codes* $\mathcal{C}, \mathcal{D} \subset \mathbb{F}_q^{m\times m}$ *with one-dimensional hulls and generators of these hulls with separable characteristic polynomials, the average complexity of finding* $P \in \mathbb{F}_q^{m\times m}$ *such that* $\mathcal{D} = P\mathcal{C}P^{-1}$ *using Algorithms 4 and 5 is*

$$\tilde{\mathcal{O}}(km^{2\omega} q^{3\sqrt{m}/(\sqrt{\log m}\log q)}),$$

which is negligible compared to $q^{m/2}$.

Proof. The smallest field extension $\mathbb{F}_{q'}$ over which the matrix U is diagonalizable is the splitting field of its characteristic polynomial, which has degree m. The average degree d of the splitting field of a monic polynomial of degree m over $\mathbb{F}_q$ verifies [12, Thm. 2]

$$d = \exp\left(C\sqrt{m/\log(m)} + O\left(\sqrt{m}\log(\log m)/\log(m)\right)\right)$$

where $C < 3$. This shows that $d = \mathcal{O}(q^{3\sqrt{m}/(\sqrt{\log m}\log q)}) \ll q^{m/2}$. We can do all the computations over $\mathbb{F}_{q^d}$, which means the number of $\mathbb{F}_q$-operations will be that of $\mathbb{F}_{q^d}$-operations multiplied by $\tilde{\mathcal{O}}(d)$ (using FFT-based algorithm for polynomial arithmetic, see for instance [13, Thm. 8.23]). Diagonalizing U and V is done in time $\mathcal{O}(m^\omega)$. Computing the subspaces $\mathcal{C} \cap E_\Lambda$, $\mathcal{D} \cap E_\Lambda$ is just linear algebra and requires $\mathcal{O}(km^{2\omega})$ operations in $\mathbb{F}_q$. Solving the system of equations takes $\mathcal{O}(m)$ multiplications in $\mathbb{F}_{q^d}$. In total, the complexity is $\tilde{\mathcal{O}}(dkm^{2\omega})$. ☐

5 Recovering Q once we know P

Note first that, given a code $\mathcal{C}_A$, the probability that a random code $\mathcal{D}$ is a conjugate of $\mathcal{C}_A$ is less than

$$|\operatorname{GL}_m(\mathbb{F}_q)|/|\operatorname{Gr}_k(\mathbb{F}_q^{m\times m})| \sim q^{m^2-k(m^2-k)}.$$

This is less than q^{-m^2} for any $m \geqslant 3$ and $2 \leqslant k \leqslant m^2-2$. Thus, it is highly unlikely that we find matrices A, B, P such that $\mathcal{D}_B = P\mathcal{C}_A P^{-1}$ without there existing a matrix Q such that $\mathcal{D} = P\mathcal{C}Q^{-1}$, and on average, the first (A, B, P) found will be correct.

The problem we are now trying to solve is the following: given two k-dimensional codes $\mathcal{C}, \mathcal{D} \subset \mathbb{F}_q^{m\times n}$, find a matrix $Q \in \operatorname{GL}_m(\mathbb{F}_q)$ such that $\mathcal{D} = \mathcal{C}Q$. Let $(C_1, \ldots, C_k)$ be a basis of $\mathcal{C}$. Given any invertible matrix $Q \in \mathbb{F}_q^{n\times n}$ such that $C_1Q, \ldots, C_kQ \in \mathcal{D}$, we have $\mathcal{C}Q = \mathcal{D}$. Define the linear map

$$\begin{aligned}\psi_{\mathcal{C}} \colon \mathbb{F}_q^{n\times n} &\longrightarrow (\mathbb{F}_q^{m\times n})^k\\ Q &\longmapsto (C_1Q, \ldots, C_kQ).\end{aligned}$$

The suitable matrices Q are exactly the elements of $\psi_{\mathcal{C}}^{-1}(\mathcal{D}^k) \cap \operatorname{GL}_n(\mathbb{F}_q)$. Concretely, computing the space $\psi_{\mathcal{C}}^{-1}(\mathcal{D}^k)$ requires $\mathcal{O}(n^2 \cdot (mnk)^{\omega-1})$ operations in

$\mathbb{F}_q$. Then, an invertible matrix Q is generally found quite easily by picking a random element in this space. Note that it may happen that invertible elements are rare in such a space. However, we claim that this situation is rather unlikely to happen. Moreover, even in the worst cases, the problem of finding such a Q can be done in polynomial time as explained in [7, Thm. 3.7] and [11].

References

1. Baldi, M., Barenghi, A., Beckwith, L., Biasse, J.F., Esser, A., Gaj, K., Mohajerani, K., Pelosi, G., Persichetti, E., O. Saarinen, M.J., Santini, P., Wallace, R.: LESS (Linear Equivalence Signature Scheme). NIST, Post–Quantum Cryptography : Additional Digital Signature Schemes (2023), https://www.less-project.com/
2. Beullens, W.: Not enough LESS: An improved algorithm for solving code equivalence problems over $\mathbb{F}_q$. In: Dunkelman, O., Jacobson, Jr., M.J., O'Flynn, C. (eds.) Selected Areas in Cryptography. pp. 387–403. Springer International Publishing, Cham (2021)
3. Beullens, W.: Graph-theoretic algorithms for the alternating trilinear form equivalence problem. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023. pp. 101–126. Springer Nature Switzerland, Cham (2023)
4. Bläser, M., Duong, D.H., Narayanan, A.K., Plantard, T., Qiao, Y., Sipasseuth, A., Tang, G.: ALTEQ. NIST, Post–Quantum Cryptography : Additional Digital Signature Schemes (2023), https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/ALTEQ-Spec-web.pdf
5. Bos, J.W., Bronchain, O., Ducas, L., Fehr, S., Huang, Y.H., Pornin, T., Postlethwaite, E.W., Prest, T., Pulles, L.N., van Woerden, W.: Hawk: a signature scheme inspired by the lattice isomorphism problem. NIST, Post–Quantum Cryptography : Additional Digital Signature Schemes (2023), https://hawk-sign.info/
6. Bouillaguet, C., Fouque, P.A., Véber, A.: Graph-theoretic algorithms for the "Isomorphism of Polynomials" problem. In: Johansson, T., Nguyen, P.Q. (eds.) Advances in Cryptology – EUROCRYPT 2013. pp. 211–227. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
7. Brooksbank, P.A., Wilson, J.B., O'Brien, E.A.: Testing isomorphism of graded algebras. Trans. Amer. Math. Soc. **372**(11), 8067–8090 (2019)
8. Caldero, P., Germoni, J.: Nouvelles histoires hédonistes de groupes et de géométries. Mathématiques en devenir, Calvage et Mounet (Feb 2018), https://hal.science/hal-02161089
9. Chou, T., Niederhagen, R., Persichetti, E., Ran, L., Randrianarisoa, T.H., Reijnders, K., Samardjiska, S., Trimoska, M.: MEDS (Matrix Equivalence Digital Signature Scheme). NIST, Post–Quantum Cryptography : Additional Digital Signature Schemes (2023), https://www.meds-pqc.org/
10. Chou, T., Niederhagen, R., Persichetti, E., Randrianarisoa, T.H., Reijnders, K., Samardjiska, S., Trimoska, M.: Take your MEDS: Digital signatures from matrix code equivalence. In: El Mrabet, N., De Feo, L., Duquesne, S. (eds.) Progress in Cryptology - AFRICACRYPT 2023. pp. 28–52. Springer Nature Switzerland, Cham (2023)
11. Couvreur, A., Debris-Alazard, T., Gaborit, P.: On the hardness of code equivalence problems in rank metric (Nov 2020), https://hal.archives-ouvertes.fr/hal-02997801, working paper or preprint

12. Dixon, J.D., Panario, D.: The degree of the splitting field of a random polynomial over a finite field. The Electronic Journal of Combinatorics pp. R70–R70 (2004)
13. von zur Gathen, J., Gerhard, J.: Modern Computer Algebra. Cambridge University Press, 3rd edn. (2013)
14. Grochow, J., Qiao, Y.: On the complexity of isomorphism problems for tensors, groups, and polynomials I: tensor isomorphism-completeness. SIAM J. Comput. **52**(2), 568–617 (2023)
15. Grove, L.C.: Classical groups and geometric algebra, vol. 39. American Mathematical Soc. (2002)
16. Kerr, M.: Algebra I Lecture Notes, https://www.math.wustl.edu/ matkerr/5031/
17. Leon, J.: Computing automorphism groups of error-correcting codes. IEEE Trans. Inform. Theory **28**(3), 496–511 (1982)
18. López, A., Maisner, D., Nart, E., Xarles, X.: Orbits of Galois Invariant n-Sets of $\mathbb{P}^1$ under the Action of PGL_2. Finite Fields Appl. **8**(2), 193–206 (2002). https://doi.org/https://doi.org/10.1006/ffta.2001.0335, https://www.sciencedirect.com/science/article/pii/S1071579701903351
19. Matsumoto, T., Imai, H.: Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In: Advances in Cryptology—EUROCRYPT'88: Workshop on the Theory and Application of Cryptographic Techniques Davos, Switzerland, May 25–27, 1988 Proceedings 7. pp. 419–453. Springer (1988)
20. Narayanan, A.K., Qiao, Y., Tang, G.: Algorithms for matrix code and alternating trilinear form equivalences via new isomorphism invariants. In: Joye, M., Leander, G. (eds.) Advances in Cryptology – EUROCRYPT 2024. pp. 160–187. Springer Nature Switzerland, Cham (2024)
21. Neiger, V., Pernet, C.: Deterministic computation of the characteristic polynomial in the time of matrix multiplication. J. Complexity **67**, 101572 (2021). https://doi.org/https://doi.org/10.1016/j.jco.2021.101572, https://www.sciencedirect.com/science/article/pii/S0885064X21000273
22. Patarin, J.: Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. In: Maurer, U.M. (ed.) Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding. Lecture Notes Comput. Sci., vol. 1070, pp. 33–48. Springer (1996). https://doi.org/10.1007/3-540-68339-9"'4, https://doi.org/10.1007/3-540-68339-9_4
23. Perret, M.: On the number of points of some varieties over finite fields. Bull. Lond. Math. Soc. **35**(3), 309–320 (05 2003). https://doi.org/10.1112/S0024609302001820, https://doi.org/10.1112/S0024609302001820
24. Ran, L., Samardjiska, S., Trimoska, M.: Algebraic algorithm for the alternating trilinear form equivalence problem. In: Esser, A., Santini, P. (eds.) Code-Based Cryptography. pp. 84–103. Springer Nature Switzerland, Cham (2023)
25. Sendrier, N.: On the dimension of the hull. SIAM J. Discrete Math. **10**(2), 282–293 (1997). https://doi.org/10.1137/S0895480195294027, https://doi.org/10.1137/S0895480195294027
26. Sendrier, N.: Finding the permutation between equivalent linear codes: The support splitting algorithm. IEEE, Trans. Inform. Theory **46**(4), 1193–1203 (2000)
27. Sendrier, N., Simos, D.E.: The hardness of code equivalence over and its application to code-based cryptography. In: Post-Quantum Cryptography 2013. Lecture Notes Comput. Sci., vol. 7932, pp. 203–216. Springer (2013)

28. Serre, J.P.: Lettre à M. Tsfasman. Astérisque **198**, 199–200 (1989)
29. Tang, G., Duong, D.H., Joux, A., Plantard, T., Qiao, Y., Susilo, W.: Practical post-quantum signature schemes from isomorphism problems of trilinear forms. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology – EUROCRYPT 2022. pp. 582–612. Springer International Publishing, Cham (2022)

A Normalizing matrices and characteristic polynomials

We consider the map introduced in Section 3.2

$$\begin{aligned}\{U \in \mathbb{F}_q^{m\times m} \mid \mathrm{Tr}(U) = \mathrm{Tr}(U^2) = 0\}/\mathbb{F}_q^\times &\longrightarrow (\mathbb{F}_q^{m-2} - (\mathbb{F}_q^{m-1} \times \{0\}))/\mathbb{F}_q^\times \\ U &\longmapsto (a_{m-3}, \dots, a_0)\end{aligned}$$

where the characteristic polynomial of U is $t^m + a_{m-3}t^{m-3} + \cdots + a_1 + a_0$ and $\mathbb{F}_q^\times$ acts on $\mathbb{F}_q^{m-2}$ via $\lambda \diamond (a_{m-3}, \dots, a_0) = (\lambda^3 a_{m-3}, \dots, \lambda^m a_0)$.

Lemma 13. *The set* $(\mathbb{F}_q^{m-2} - (\mathbb{F}_q^{m-1} \times \{0\}))/\mathbb{F}_q^\times$ *has*

$$q + q^2 + \cdots + q^{m-3} \sim q^{m-3}$$

elements.

Proof. The set $(\mathbb{F}_q^{m-2}/\mathbb{F}_q^\times) - \{(0, \dots, 0)\}$ is a subset of the set of $\mathbb{F}_q$-rational points of the weighted projective space $\mathbb{P}^{m-3}_{3,\dots,m}$ of dimension $m-3$ and weights $3, \dots, m$ over $\mathbb{F}_q$ [23, Lem. 6]. By [23, Lem. 7], this has $(q^{m-2} - q)/(q-1)$ elements. □

Lemma 14. *The subset* $\mathrm{Sep}_{q,m} \subset (\mathbb{F}_q^{m-2} - (\mathbb{F}_q^{m-1} \times \{0\}))/\mathbb{F}_q^\times$ *of classes of separable polynomials has* $\sim q^{m-3}$ *elements.*

Proof. The monic inseparable polynomials of degree m over $\mathbb{F}_q$ are the points of an open subset of a hypersurface of degree $2m-2$ in $\mathbb{P}^m$ [18, §1]. The set of inseparable polynomials whose coefficients of degree $m-1, m-2$ vanish is the intersection of this with two hyperplanes that do not contain it. Hence, it is an open subset of a hypersurface of degree $\leqslant 2m-2$ in $\mathbb{P}^{m-2}$, and has

$$\mathcal{O}\left((2m-2)q^{m-3}\right)$$

elements by the Serre bound [28, Théorème]. Since every element of $\mathbb{P}^{m-3}_{3,\dots,m}(\mathbb{F}_q)$ has exactly $q-1$ preimages in $\mathbb{F}_q^{m-2}$ [23, Lem. 7], this means that there are $\mathcal{O}(mq^{m-4})$ classes of inseparable polynomials in $\mathbb{P}^{m-3}_{3,\dots,m}(\mathbb{F}_q)$. Hence, by Lemma 13, $\mathrm{Sep}_{q,m}$ has $q^{m-3} - \mathcal{O}(mq^{m-4}) \sim q^{m-3}$ elements. □

Here is how to choose and compute a normalized representative of any element $\chi = (a_{m-3}, \dots, a_0) \in \mathbb{F}_q^{m-2}$ modulo $\mathbb{F}_q^\times$. First, the normalized representative of 0 is itself. Now, consider $\chi \in \mathbb{F}_q^{m-2} - \{0\}$. Denote by $i_0 < i_1 < \cdots < i_\ell$ the indices such that $a_{m-i_j} \neq 0$. Choose a generator g of $\mathbb{F}_q^\times$, and write $a_{m-i_j} = g^{s_j}$.

– If i_0 is prime to $q-1$, there is a unique $\lambda \in \mathbb{F}_q^\times$ such that $\lambda^{i_0} a_{m-i_0} = 1$; this λ is $g^{-s_0 \cdot i_0^{-1} \bmod q}$. In that case, we choose $\chi' = \lambda \diamond \chi$ to be the normalized representative of χ.
– If $d_0 \stackrel{\text{def}}{=} \gcd(i_0, q-1) > 1$, there are d_0 elements λ satisfying this property. Let us describe how to find the right one.
  1. Here is how to compute one such λ. Write $i_0 = d_0 i'_0$, and denote by j'_0 the inverse of i'_0 modulo $q-1$. The set $\mathbb{F}_q^\times / (\mathbb{F}_q^\times)^{i_0}$ has d_0 elements: the equivalence classes of $1, g, \ldots, g^{d_0-1}$. Compute the Euclidean division $s_0 = s'_0 \cdot d_0 + r_0$ of s_0 by d_0. Then the element $\lambda_0 \stackrel{\text{def}}{=} g^{-s'_0 j'_0}$ satisfies $\lambda_0^{i_0} a_{m-i_0} = g^{r_0}$. Any product of λ_0 by a d_0-th root of unity in $\mathbb{F}_q$ still satisfies this relation.
  2. Now let a_{m-i_1} be the next nonzero coefficient of χ. We want to normalize $a_{m-i_1}\lambda_0^{i_1} = g^{s_1}$ by multiplying it by a d_0-th root of unity. Set $d_1 = \gcd(d_0, i_1)$. For any integer δ, denote by $\mu_\delta(\mathbb{F}_q)$ the group of δ-th roots of unity in $\mathbb{F}_q$. The set
  $$\mathbb{F}_q^\times / \mu_{d_0}(\mathbb{F}_q)^{i_1} = \mathbb{F}_q^\times / \mu_{d_1/d_0}(\mathbb{F}_q)$$
  has $(q-1)d_1/d_0$ elements. Write $i_1 = d_1 i'_1$ and denote by j'_1 the inverse of i'_1 modulo d_0. Compute the Euclidean division $s_1 = s'_1 \cdot d_1(q-1)/d_0 + r_1$. The element $\alpha_1 = g^{-s'_1 j'_1 (q-1)/d_0}$ satisfies $\alpha_1^{i_1} g^{s_1} = g^{r_1}$. Set $\lambda_1 \stackrel{\text{def}}{=} \lambda_0 \alpha_1$. We have $\lambda_1^{i_0} a_{m-i_0} = g^{r_0}$ and $\lambda_1^{i_1} a_{m-i_1} = g^{r_1}$.
  3. If $d_1 \neq 1$, continue with $d_2 = \gcd(d_1, i_2)$. Stop after the k-th step if $k = \ell$ or $d_k = 1$. Return $\chi' = \lambda_k \cdot \chi$. The algorithm is summed up below.

The element $\chi' \in \mathbb{F}_q^{m-2}$ returned by this algorithm is equivalent to χ. More generally, given equivalent inputs in $\mathbb{F}_q^{m-2}$, it returns the same output.

Algorithm 6: NORMALIZE

Data: Matrix $U \in \mathbb{F}_q^{m \times m}$, tuple $\chi = (a_{m-3}, \ldots, a_0) \in \mathbb{F}_q^{m-2} - \{0\}$
Generator g of $\mathbb{F}_q^\times$
Result: Matrix $U' \in \mathbb{F}_q^{m \times m}$, tuple $\chi' \in \mathbb{F}_q^{m-2} - \{0\}$

Set $d = q-1$, $i = 2$ and $\lambda = 1$
while $d \neq 1 \wedge (a_{m-i-1}, \ldots, a_1) \neq (0, \ldots, 0)$ **do**
  Set $i = \min\{j \in \{i+1 \ldots m\} \mid a_{m-j} \neq 0\}$
  Parse $\lambda^i a_{m-i} = g^s$
  Set $d' = \gcd(d, i)$ and $i' = i/d'$
  Compute inverse $j' \in \mathbb{Z}$ of i' modulo d
  Compute Euclidean division $s = s' \cdot (q-1)d'/d + r$
  Set $\lambda \leftarrow \lambda g^{-s'j'(q-1)/d}$
  Set $d \leftarrow d'$, $i \leftarrow i+1$
return λ

Remark 15. The complexity of Algorithm 6 is $\mathcal{O}(m \log(q))$. To reduce its impact on the complexity of our attack, one can precompute the normalization of the most frequent characteristic polynomials. For instance, on may compute in advance the suitable λ for vectors $(a_{m-3}, \dots, a_0)$ such that $a_{m-3}, a_{m-4} \neq 0$. Since 3 and 4 are coprime, a unique λ is found knowing only a_{m-3}, a_{m-4}: constructing a dictionary $\{(a_{m-3}, a_{m-4}) : \lambda\}$ allows to precompute the normalization for $(q-1)^2 q^{m-4} \sim q^{m-2}$ elements of $\mathbb{F}_q^{m-2}$, that is, almost all of them.

B Proportion of codes with one-dimensional hull

This section explains how the following proposition can be deduced from a similar statement found in the literature.

Proposition 1. *The proportion of* $m \times m$ *matrix codes contained in* $\ker(\mathrm{Tr})$ *and whose hull has dimension 1 is asymptotically equal to*

$$\frac{1}{q}\left(1 + \mathcal{O}\left(\frac{m^2}{q^{(m^2-1)/2}}\right)\right).$$

Sendrier's work [25] gives detailed results about the number of codes with a hull of given dimension. His results are proven in the case of codes inside $\mathbb{F}_q^n$ with the usual inner product. In our case however, we consider $\mathbb{F}_q^{m \times m}$ endowed with the bilinear form $(X, Y) \mapsto \mathrm{Tr}(XY)$. Denoting by $\sigma_{n,i}$ the number of totally isotropic $[n, k]_q$-codes for a given bilinear form, the number $A_{n,k,1}$-codes whose intersection with their orthogonal complement has dimension 1 is equal to [25, Theorem 2]

$$A_{n,k,1} = \sum_{i=1}^{k} \begin{bmatrix} n-2i \\ k-i \end{bmatrix} \begin{bmatrix} i \\ 1 \end{bmatrix} (-1)^{i-1} q^{(i-1)(i-2)/2} \sigma_{n,i}$$

where $\begin{bmatrix} n \\ k \end{bmatrix}$ is the *Gaussian binomial coefficient* which denotes the number of k-dimensional linear subspaces of $\mathbb{F}_q^n$. The proof of this result does not involve the nature of the considered non-degenerate bilinear form. Sendrier goes on to show [25, Theorem 3], using asymptotic results based on explicit values of $\sigma_{n,i}$ specific to a bilinear form of discriminant 1, that for $1 \leqslant k \leqslant n/2$,

$$A_{n,k,1} q^{k(k+1)/2} = \left(\begin{bmatrix} n \\ k \end{bmatrix} \begin{bmatrix} k \\ 1 \end{bmatrix} \prod_{i=0}^{k-1} (q^i - (i \mod 2)) \right) \left(1 + \mathcal{O}\left(\frac{k}{q^{n/2-1}}\right)\right).$$

There are different formulas of $\sigma_{n,i}$ given in [25, Theorem 1] depending on the remainders of n, q modulo 2 and 4 and on the size of k. However, they are asymptotically equivalent, which yields this uniform result. For a bilinear form of different discriminant, these formulas are simply permuted; a general expression may be found in [8, IV, Proposition 3.5]. This does not change the asymptotic result above. Moreover, for $k \geqslant n/2$, the number of $[n, k]$-codes with one-dimensional hull is that of $[n, n-k]$-codes with one-dimensional hull, since the hull of a code

$\mathcal{C}$ is exactly that of its dual. In particular, this means that for any k such that $1 \leqslant k \leqslant n-1$,

$$\frac{A_{n,k,1}}{\begin{bmatrix} n \\ k \end{bmatrix}} = \frac{1}{q}\left(1 + \mathcal{O}\left(\frac{\min(k, n-k)}{q^{n/2-1}}\right)\right)$$

or equivalently, that the proportion of codes whose hull has dimension 1 is asymptotically equivalent to $1/q$.