DIMACS Series in Discrete Mathematics
and Theoretical Computer Science
Volume **00**, 19xx

Constructive Recognition of a Black Box Group Isomorphic to $GL(n, 2)$

Gene Cooperman, Larry Finkelstein, and Steve Linton

Abstract. A Monte Carlo algorithm is presented for constructing the natural representation of a group G that is known to be isomorphic to $GL(n, 2)$. The complexity parameters are the natural dimension n and the storage space required to represent an element of G. What is surprising about this result is that both the data structure used to compute the isomorphism and each invocation of the isomorphism require polynomial time complexity. The ultimate goal is to eventually extend this result to the larger question of constructing the natural representation of classical groups. Extensions of the methods developed in this paper are discussed as well as open questions.

1. Introduction

The principal objective of this paper is a demonstration of the feasibility of obtaining the natural (projective) matrix representation for a classical group initially presented as a *black box group*. In this model, group elements are encoded by binary strings of uniform length N, and group operations are performed by an oracle (the black box). The oracle can compute the product of elements, the inverse and recognize the identity element in time polynomial in N. The most important examples of black box groups are matrix groups and permutation groups. This effort was initially motivated by recent efforts to identify the structure of matrix groups defined over finite fields (see [**3**] [**7**] and [**9**] in this volume). However, an important advantage of working in a black box setting is that the techniques rely only on the structure of the group and not on the representation.

Our major result is the following.

Theorem 1.1. *Let G be a black box group specified by a generating set $\mathcal{G}$ which is known to be isomorphic to $GL(n, 2)$ for some n. Let M be a known upper bound on n, let μ be the time required to perform a group operation in G, let ρ be the time required to compute a (nearly) uniform random element of G, and let ϵ be the time for a field operation in $GF(2)$. Then a Monte Carlo algorithm exists which in time $O((M \log^2 M + n^2)\mu + M\rho + n^3\epsilon)$ can determine the value of n and produce a data*

1991 *Mathematics Subject Classification.* Primary 20G40, 20-04; Secondary 20C40.
The first two authors were partially supported by NSF Grant CCR-9509783.

structure which can then be used to compute an isomorphism $\Theta : G \longrightarrow GL(n, 2)$*. Each invocation* $\Theta(x)$*,* $x \in G$*, takes time* $O(n^2\mu)$*.*

In the case where elements of G are given by binary strings of length N, $2^{(n-1)^2} \leq 2^{n(n-1)/2} \prod_{i=1}^{i} (2^n - 1) = |GL(n, 2)| \leq 2^N$. Hence we can choose $M = 1 + \sqrt{N}$. If G is specified as a subgroup of $GL(m, q)$, then $M = m + 1$ suffices by Lemma 4.5.

In this paper, we will need to make use of a function $\mathtt{Rand}(\mathcal{G})$ which has as input the generating set $\mathcal{G}$ for G and returns a randomly chosen element of G. Ideally, one would like to have an efficient implementation of $\mathtt{Rand}()$ which produces elements which are uniformly distributed. In the case where G is black box group, Babai [**1**] achieves this goal. Simply put, he shows how to construct from $\mathcal{G}$, a set of $O(\log |G|)$ elements, at a cost of $O(\log^5 |G|)$ group multiplications, from which *nearly uniform* distributed random elements of G can be obtained at a cost of $O(\log^4 |G|)$ group multiplications per random element. Although this result is of significant theoretical interest, it does not appear suitable for practical implementations at this point. In this situation, heuristic methods for generating nearly uniformly distributed group random elements are used. A novel heuristic for such random elements is described in [**4**] and has been used by several authors (see [**9**] in this volume).

It is also required that we know the prime factorization for each divisor of $|G|$ of the form $2^i - 1$, $1 \leq i \leq n$. This allows us to use the Bounded Order Algorithm of Celler and Leedham-Green [**5**] for computing the order of an element x of G when it is known in advance that $|x|$ divides $2^i - 1$, $1 \leq i \leq n$.

Our approach is to determine elements of G which can easily be found in a black box setting and which relate naturally to the action of G on an n-dimensional vector space V defined over $GF(2)$. An obvious choice is the set of transvections of G and an associated block system of imprimitivity formed from the action of G on the set of transvections.

The conjugation action of G on this block system (there actually are two of them, but they are conjugate in $Aut(GL(n, 2))$) is permutation equivalent to the action of G on the non-zero vectors of V. The difficulty lies in describing this action explicitly. What is surprising about our result is not the initial approach, which in its own right has been used extensively by other authors, but that the data structures for computing Θ can be determined without having to enumerate the elements of the block system which would then require time exponential in n.

Theorem 1.1 is a preliminary result. The choice of $GL(n, 2)$ was made to remove certain technical issues which occur with arbitrary finite fields but of sufficient complexity to reveal the difficulties in obtaining a more general result. The only component procedure of our algorithm which is Monte Carlo and not Las Vegas is the determination of n. Removing this obstruction within the required time bound is of considerable interest. This issue along with possible generalizations are discussed in the the last section.

2. Overview of the Algorithm

Transvections play a key role both in determining the value of n and in creating the data structures required to compute the isomorphism $\Theta : G \longrightarrow GL(n, 2)$. Each transvection $\tau_{v,f}$ of $GL(n, 2)$ can be identified with a unique vector $v \in V$ and a linear functional $f \in V^*$ such that $f(v) = 0$ and $\tau_{v,f}$ acts on V according to the following rule:

$$w\tau_{v,f} = w + f(w)v,\ w \in V.$$

We refer to v as the *center* of the transvection $\tau_{v,f}$.

The construction of Θ depends on three critical properties of transvections.

(1): Given two transvections $\tau_{v,f}$ and $\tau_{u,g}$ of $GL(n,2)$, one can determine certain relations between u, v, f, and g from computing the order of their product. For example, if $|\tau_{v,f}\tau_{u,g}| = 4$, then $(\tau_{v,f}\tau_{u,g})^2$ is a transvection and either $f(u) = 0$, $g(v) = 1$ and $(\tau_{v,f}\tau_{u,g})^2 = \tau_{u,f}$ or $f(v) = 1$, $g(v) = 0$ and $(\tau_{v,f}\tau_{u,g})^2 = \tau_{v,g}$ (Lemma 5.2).

(2): The permutation representation of $GL(n,2)$ on the set of all transvections is imprimitive with precisely two maximal systems of imprimitivity. The representatives of these block systems which contain $\tau_{v,f}$ can be described as the sets $\Gamma(f) = \{\tau_{x,f} : x \in V, f(x) = 0, v \neq 0\}$ and $\beta(v) = \{\tau_{v,h} : x \in V, h(v) = 0, h \neq 0\}$.

(3): If $\tau_{v,f}$ is a transvection and α is an arbitrary element of $GL(n,2)$, then

$$\tau_{v,f}^{\alpha} = \tau_{v\alpha, f\alpha^*}$$

where $\alpha^* \in GL(V^*)$ is defined by $(f\alpha^*)(w) = f(w\alpha^{-1})$, for $w \in V$.

The first step in the algorithm is to determine the value of n and identify an element $t \in G$ that will map under Θ to a transvection of $GL(n,2)$. This is rather straightforward and is discussed in section 4. Under the assumption that Θ is an isomorphism, we may set $\Theta(t) = \tau_{v,f}$ for some $v \in V$ and $f \in V^*$ with $f(v) = 0$. Note that since transvections form a single conjugacy class in $GL(n,2)$, the choice of v and f is arbitrary subject only to $f(v) = 0$.

It will be convenient to denote the preimage of a transvection $\tau_{u,g}$ under Θ by $t_{u,g}$ and refer to elements of this conjugacy class of G as transvections as well. Furthermore, we will often denote an arbitrary element of this conjugacy class of G in the form $t_{w,h}$. This is legitimate as long as we don't make any assumptions on the values of w and h that can't be properly inferred from Θ. For example, if $t_{w,h} = t_{u_1,g}t_{u_2,g}$ and we have already extended Θ to $t_{u_1,g}$ and $t_{u_2,g}$, then $\Theta(t_{w,h}) = \Theta(t_{u_1,g})\Theta(t_{u_2,g}) = \tau(u_1,g)\tau(u_2,g) = \tau(u_1+u_2,g)$. Thus $w = u_1 + u_2$ and $h = g$.

Given a transvection $t_{v,f}$ of G, we may successively use properties (1) and (2) to construct subsets B_Γ and B_β of G which we will refer to as a *dual block pair* and which satisfies the following properties:

- $\Theta(B_\Gamma) = \{\tau_{v_1,f}, \tau_{v_2,f}, \ldots, \tau_{v_{n-1},f}\}$ where $\{v_1, \ldots, v_{n-1}\}$ is a basis for $\Gamma(f)$.
- $\Theta(B_\beta) = \{\tau_{v_n,g_1}, \ldots, \tau_{v_n,g_{n-1}}\}$ where $f(v_n) = 1$ and $\{g_1, \ldots, g_{n-1}\}$ is a basis for $\beta(v_n)$.
- $g_i(v_j) = \delta_{ij}$, $1 \leq i,j \leq n-1$, where δ_{ij} is interpreted as an element of $GF(2)$.

The construction of B_Γ and B_β is presented in section 6. Here $t_{v,f} = t_{v_1,f}$. We set $B_\Gamma = \{t_{v_1,f}, \ldots, t_{v_{n-1},f}\}$ and $B_\beta = \{t_{v_n,g_1}, \ldots, t_{v_n,g_{n-1}}\}$.

Since $f(v_n) = 1$, the set $\{v_1, v_2, \ldots, v_n\}$ is a basis of V. Assuming we know the restriction of Θ to $B_\Gamma \cup B_\beta$, the image $\Theta(x)$, for an arbitrary $x \in G$, is then determined by computing the coefficients of $v_i\Theta(x)$ relative to $\{v_1, v_2, \ldots, v_n\}$ for $1 \leq i \leq n$. To do this, we make use of (3) above. Thus, for $1 \leq i \leq n-1$,

$$\tau_{v_i,f}^{\Theta(x)} = \tau_{v_i\Theta(x), f\Theta(x)^*} = \tau_{w_i,h_i},$$

and

$$\tau_{v_n,g_{n-1}}^{\Theta(x)} = \tau_{v_n\Theta(x),g_{n-1}\Theta(x)^*} = \tau_{w_n,h_n}.$$

Thus the problem of expressing $v_i\Theta(x)$ as a linear combination of $\{v_1, v_2, \ldots, v_n\}$ reduces to the following problem. Given the transvection τ_{w_i,h_i}, express w_i as a linear combination of $\{v_1, v_2, \ldots, v_n\}$, $1 \leq i \leq n$. Since we do not have any explicit way of constructing w_i, this does not appear to be all that useful. However, for $1 \leq i \leq n-1$, $\tau_{w_i,h_i} = \Theta(t_{v_i,f}^x)$ and so the preimage of τ_{w_i,h_i} under Θ is precisely $t_{v_i,f}^x = t_{w_i,h_i}$ which can be computed in G.

More generally, assume we are given a transvection $t_{w,h}$ of G and we want to express $w = \sum_{i=1}^n a_i v_i$. Initially, we have no information on the values of w and h. The solution of this problem occurs in two stages and is presented in section 7. In the first stage, we reduce to the case where $h = f$ and simultaneously determine the coefficient a_n. This reduction is fairly straightforward and makes use of properties of transvections developed in section 5. In the second stage, we know that $t_{w,f} \in \langle B_\Gamma \rangle$ and want to express $t_{w,f}$ in terms of the basis B_Γ for $\langle B_\Gamma \rangle$. This means, finding coefficients $\{a_1, \ldots, a_{n-1}\}$ such that

$$t_{w,f} = t_{v_1,f}^{a_1} \cdots t_{v_{n-1},f}^{a_{n-1}}.$$

In the black box setting, we do not have an explicit value for w and so the usual linear algebra methods do not work. Instead, we use the existence of our dual block pair (B_Γ, B_β) which allows us to easily find explicit values for $\{a_1, \ldots, a_{n-1}\}$. The details are presented in section 7.

3. The Top Level Procedures

The procedure `Construct-Data-Structures` is used to set up the data structures required for each call to Θ. The input is the generating set $\mathcal{G}$. We will not be concerned with precise probability estimates. Rather, each of the procedures that is called will be responsible for returning the correct answer with constant probability, i.e. with probability at least c for some fixed c, $0 < c < 1$. Each procedure will return `false` if it can detect that the returned value is incorrect. In general, each procedure can achieve higher reliability in the standard way by sampling more random chosen elements. We initialize M to be an upper bound on the dimension n of G. As noted in the introduction, if G is a black box group specified by binary strings of length N, then M may be set to $1 + \sqrt{N}$.

```
Procedure Construct-Data-Structures($\mathcal{G}$)
    Input: A generating set $\mathcal{G}$ for $G$.
    Output: The dimension $n$ of $G$, a transvection $t = t_{v,f} \in G$, an element $\sigma \in G$
        which commutes with $t$ and whose order contains a primitive prime divisor
        of $2^{n-2} - 1$ and a dual block pair $(B_\Gamma, B_\beta)$. (If $n - 2 = 6$, then we require
        that $|\sigma| = 63$.)
    Reliability: 1/128
    Complexity: $O((M \log^2 M + n^2)\mu + M\rho + n^3\epsilon)$ where $\mu, \rho, \epsilon$ are as defined in
        Theorem 1.1.
If $\mathcal{G} \subseteq GL(m, q)$
        Set $M \leftarrow m + 1$
                [See Lemma 4.5.]
Else If $G$ is a Black Box group
        Let $N$ be the binary string length for specifying elements of $G$
```

```
          Set $M \leftarrow \lfloor 1+\sqrt{N} \rfloor$
Set $(\ell, t, \sigma) \leftarrow$ Find-Element-in-$\mathcal{C}_2(\mathcal{G}, M)$
          [See section 4]
If $\ell = 0$
          Return(false)
Set $n = \ell + 2$
Set $B_\Gamma \leftarrow$ Construct-Block-Basis$(n, t, \sigma, \mathcal{G})$
          [See section 6]
If $B_\Gamma =$ false
          Return(false)
Set $B_\beta \leftarrow$ Construct-Dual-Block-Pair $(n, t, \sigma, B_\Gamma, \mathcal{G})$
          [See section 6]
If $B_\beta =$ false
          Return(false)
```

Construct-Data-Structures is Monte Carlo with constant reliability. The procedure Find-Element-in-$\mathcal{C}_2$ has constant reliability that the returned values (ℓ, t, σ) are correct, hence may return the wrong answer without it being detected. It should be noted, that if $n = \ell + 2$ is correct, then it is certain that t is a transvection of G and σ is a $ppd(n, 2, n-2)$ element of G. The two other procedures Construct-Block-Basis and Construct-Dual-Block-Pair are both Las Vegas given that n has been correctly determined.

We can now specify the isomorphism $\Theta\colon G \longrightarrow GL(n, 2)$. Elements of G will be represented as $n \times n$ matrices over $GF(2)$ relative to the basis for $V = \{v_1, \dots, v_n\}$ determined by the dual block pair (B_Γ, B_β), where $B_\Gamma = \{t_{v_1,f}, \dots, t_{v_{n-1},f}\}$, with $t_{v,f} = t_{v_1,f}$, and $B_\beta = \{t_{v_n,g_1}, \dots, t_{v_n,g_{n-1}}\}$.

```
Procedure $\Theta$
     Input: An arbitrary matrix $x \in G$, $B_\Gamma$ and $B_\beta$.
     Output: A matrix $(a_{i,j})$ for $1 \le i \le n$, $1 \le j \le n$.
     Complexity:: $O(n^2\mu)$, where $\mu$ is as defined in Theorem 1.1.
For $i \leftarrow 1$ to $n-1$
          Set $(a_{i,1}, \dots, a_{i,n-1}) \leftarrow$ Compute-Transvection-Center$(t^x_{v_i,f}, B_\Gamma, B_\beta)$
Set $(a_{n,1}, \dots, a_{n,n-1}) \leftarrow$ Compute-Transvection-Center$(t^x_{v_n,g_1}, B_\Gamma, B_\beta))$
          [See section 7]
Return$((a_{i,j}))$
```

4. Finding a Transvection

In this section, we derive a method for finding transvections of $GL(n, 2)$ that does not depend on having the natural representation of $GL(n, 2)$ in hand. This can then be used as the conceptual basis for the procedure Find-Element-in-$\mathcal{C}_2$. This procedure is Monte Carlo with constant reliability and may return an incorrect answer without it being detected. However, if we are willing to spend sufficient time, then the probability of error can be reduced below any given threshold.

An element of $G = GL(n, 2)$ is *irreducible* if its characteristic polynomial is irreducible over $GF(2)$, or equivalently if it acts irreducibly on V. A Singer cycle is a cyclic subgroup of order $2^n - 1$ which acts transitively on the non-zero vectors of V. It is well known that all Singer cycles are conjugate and that each irreducible element of $GL(n,2)$ is conjugate to an element of a Singer cycle. A Singer cycle can be constructed through the embedding of $GL(1,2^n) = GF(2^n)^*$ into $GL(n,2)$. Furthermore, one can also embed $GF(2^n)^*$ extended by $\mathrm{Aut}(GF(2^n))$ into $GL(n,2)$, where $\mathrm{Aut}(GF(2^n))$ is a cyclic group of order n induced by the Frobenius automorphism. It then follows that each irreducible element contained in a Singer Cycle is conjugate to n of its powers.

Definition 4.1. *A prime p is said to be a primitive prime divisor of $2^n - 1$ if $p \mid (2^n - 1)$ but $p \not| (2^j - 1)$, $j < n$. An element σ of $GL(n,2)$ is said to be a ppd$(n,2,e)$ element, $n/2 < e \leq n$, if σ satisfies two conditions. First, it should fix an $n - e$ dimensional subspace of the underlying vector space. Second, $|\sigma| = 63$ when $e = 6$ and $|\sigma|$ contains a primitive prime divisor of $2^e - 1$ when $e \neq 6$. A ppd$(n,2,n)$ element is said to be primitive.*

Remark 4.2. *It follows from a result of Zsigmondy that for $e \neq 6$, $2^e - 1$ always has a primitive prime divisor. Thus a ppd$(n,2,e)$ element exists for all values of n and e. We employ a slightly more restrictive definition than that given by Niemeyer and Praeger in these proceedings.*

We now define a class of elements of $GL(n,2)$ that occur with high frequency and which allow us to identify transvections in a black box setting.

Definition 4.3. *Let τ be a fixed transvection and ν a ppd$(n,2,n-2)$ element of order $2^{n-2} - 1$. Denote by $\mathcal{I}_2$, the set of all elements of the form $\tau\sigma$, where $\sigma \in \langle \nu \rangle$, and σ is a ppd$(n,2,n-2)$ element. Set $\mathcal{C}_2 = \mathcal{I}_2^{GL(n,2)}$.*

Lemma 4.4. $|\mathcal{C}_2|/|GL(n,2)| \geq 1/(4(n-1))$.

Proof. Let $V = U_1 \oplus U_2$ where $U_1 = C_V(\nu)$ and $U_2 = [V,\nu]$. Then τ centralizes U_2 and stabilizes U_1. Hence, $C = C_{GL(n,2)}(\tau\nu) = C_{GL(U_1)}(\tau|U_1) \times \langle \nu \rangle \cong Z_2 \times Z_{2^{n-2}-1}$. Furthermore, $C = C_{GL(n,2)}(\tau\sigma)$ for each $\tau\sigma \in \mathcal{I}_2$. If $N = N_{GL(n,2)}(\langle \tau\nu \rangle)$, then $N/\langle \tau\nu \rangle$ is cyclic of order $n-2$. Also, two elements of $\mathcal{I}$ are conjugate in G if and only if they are conjugate in N. This follows from the structure of $C_{GL(n,2)}(\tau) = 2^{2n-3}GL(n-2,2)$. In particular, each conjugacy class contained in $\mathcal{C}_2$ contains precisely $n-2$ elements of $\mathcal{I}_2$. Thus

$$\begin{aligned} |\mathcal{C}_2| &= \{|G|/(2(2^{n-2}-1))\}|\mathcal{I}_2|/(n-2) \\ &= \{|G|/(2)\}|\mathcal{I}_2|/((2^{n-2}-1)(n-2)). \end{aligned}$$

If $n \neq 8$, then $|\mathcal{I}_2|/(2^{n-2} - 1)$ is the proportion of primitive elements in a Singer cycle of $GL(n-2,2)$, and if $n-2 = 6$, then it is the proportion of generating elements. However, it is shown in [**8**, Lemmas 2.3 and 2.4] that this number is at least $n-2/(n-1)$, except when $n-2 = 6$ in which case it is at least $n-2/(2(n-1))$. Thus

$$|\mathcal{C}_2|/|GL(n,2)| \geq 1/4(n-1)$$

as required. □

The next result will be useful in the case where the generating matrices $\mathcal{S}$ for G are given as $m \times m$ matrices over $GF(q)$.

Lemma 4.5. *If $GL(n,2)$ is a subgroup of $GL(m,q)$, then $n \leq m+1$.*

Proof. Let ζ be a Singer cycle of $GL(n,2)$. If $(q, 2^n - 1) = 1$, then in some extension field $GF(q^r)$, ζ is diagonalizable and has n distinct conjugates. This implies that ζ has at least n distinct eigenvalues as an element of $GL(m, q^r)$. Hence $n \leq m$. If $(q, 2^n - 1) \neq 1$, then q is a power of a prime distinct from 2 and so $(q, 2^{n-1} - 1) = 1$ (since $(2^n - 1) - (2^{n-1} - 1) = 2^{n-1}$). The same argument using an element of order $2^{n-1} - 1$ which generates a Singer cycle of $GL(n-1, 2)$ then yields $n - 1 \leq m$. □

In our context, G is a black box group isomorphic to $GL(n,2)$ for some unknown value of n. We need a method for characterizing elements of $\mathcal{C}_2$ of G which does not depend on any of the usual linear algebra properties.

Lemma 4.6. *Let G be a black box group which is isomorphic to $GL(n,2)$. A necessary condition for $g \in G$ to be a ppd$(n,2,e)$ element for some value of e, is that*

- $|g| \mid 2^e - 1$*, and*
- $|g|$ *contains a primitive prime divisor of* $2^e - 1$.

Lemma 4.6 leads to the following method for identifying elements of $\mathcal{C}_2$ of G.

Lemma 4.7. *Let $\mathcal{S}$ be a sequence of elements of G and suppose it is known that $\mathcal{S}$ contains an element of $\mathcal{C}_2$. Let $\mathcal{S}'$ be the subset of $\mathcal{S}$ consisting of those elements g with the property that $|g| = 2k_g$ with k_g odd. For each $g \in \mathcal{S}'$ define $t_g = g^{k_g}$ and $\sigma_g = g^2$. (Thus, $\langle g \rangle = \langle t_g \rangle \times \langle \sigma_g \rangle$.) For $g \in \mathcal{S}'$, let ℓ_g be a non-negative integer set according to the following rule. If $k_g = 63$, set $\ell_g = 6$. If k_g divides $2^e - 1$ and contains a primitive prime divisor of $2^e - 1$, set $\ell_g = e$. Otherwise, set $\ell_g = 0$. Let $\ell = max\{\ell_g : g \in \mathcal{S}'\}$. Then $\ell = n - 2$. Further, for those $g \in \mathcal{S}'$ satisfying $\ell_g = \ell$, t_g is a transvection and σ_g is a ppd$(n, 2, n-2)$ element.*

Proof. Recall from Definition 4.3 that if $g \in \mathcal{S}$ is an element of $\mathcal{C}_2$, then t_g is a transvection and σ_g is a *ppd*$(n, 2, n-2)$ element. Hence $\ell = \ell_g = n - 2$.

We claim that any element h with $\ell_h = \ell$ will be an element of $\mathcal{C}_2$. To see this, let V be an n dimensional $GF(2)$–module for G. Since σ_h commutes with the involution t_h, σ_h stabilizes $C_V(t_h)$. Furthermore, since $dim(C_V(t_h)) \geq n/2$, it follows that either $dim(C_V(t_h)) = n - 2$ or $dim(C_V(t_h)) = n - 1$. The case $dim(C_V(t_h)) = n - 2$ is impossible since in this case, $C_G(t_h)$ is isomorphic to an extension of a 2-group of order 2^{4n-12} by $GL(2,2) \times GL(n-4, 2)$ and this is incompatible with the conditions on k_h. Thus t_h is a transvection. □

The following procedure `Find-Element-Of-`$\mathcal{C}_2$`-In-Set` is based on Lemma 4.7. The input is a set $\mathcal{S}$ of elements of G and an upper bound M on the dimension of G. For example, if G is specified by $m \times m$ matrices over $GF(q)$, then we can choose $M = m + 1$ by Lemma 4.5. If it is known in advance that $\mathcal{S}$ contains an element $g \in \mathcal{C}_2$, then `Find-Element-Of-`$\mathcal{C}_2$`-In-Set` returns (ℓ_g, t_g, σ_g). If it is not known for certain that $\mathcal{S}$ contains an element of $\mathcal{C}_2$, then the procedure will find an element $g \in \mathcal{S}$ which appears most likely to be an element of $\mathcal{C}_2$ and returns (ℓ_g, t_g, σ_g). Otherwise, the procedure concludes that $\mathcal{S}$ does not contain an element of $\mathcal{C}_2$ and returns a triple (ℓ, t, s) with $\ell = 0$.

The procedure makes use of the functions `Order-Test` and `Bounded-Order`. `Order-Test` has input an element $x \in G$ and the bound M on the dimension of G. It returns either the smallest positive integer $i \leq M$ such that $|x| \mid 2^i - 1$ or else 0 if no such i exists. Using the standard doubling algorithm together with the observation that $|x| \mid 2^i - 1$ if and only if $x^{2^i} = x$, it is easy to see that `Order-Test` requires at most $O(\log M)$ multiplications. The function `Bounded-Order` has as input an element x, an integer i such that $|x| \mid 2^i - 1$ and returns $|x|$. An implementation can be based on the algorithm described in [**5**] and requires $O(\log^2 M)$ multiplications. We assume that this function has access to the prime factorization of $2^i - 1$, $i \leq M$, and do not consider this in the complexity. Similarly, for each $i \leq M$, we assume that we can test if a divisor of $2^i - 1$ is primitive.

```
Procedure Find-Element-Of-$\mathcal{C}_2$-In-Set
    Input: $(\mathcal{S}, M)$ where $\mathcal{S}$ is a set of elements of $G$ and $M$ is an upper bound on
        the dimension of $G$.
    Output: A triple $(\ell, t, \sigma)$. If the procedure can verify that $\mathcal{S}$ does not contain
        an element of $\mathcal{C}_2$, then $\ell$ is set to 0. Otherwise, $\ell$ is assigned a value believed
        to be $n - 2$, an involution $t$ and an element $\sigma$ which commutes with $t$ such
        that $g = t\sigma \in \mathcal{S}$, and $g$ is believed to be an element of $\mathcal{C}_2$.
    Complexity: $O(|\mathcal{S}| \log^2 M\mu)$ where $\mu$ is as defined in Theorem 1.1

Set $(\ell, t, \sigma) \leftarrow (0, \texttt{nil}, \texttt{nil})$
For non-trivial $g \in \mathcal{S}$ do
        Set $\sigma_g = g^2$
        Set $\ell_g \leftarrow$ Order-Test$(\sigma_g, M)$
                    $[\ell_g \leftarrow \min\{i\colon\ i \leq M, |\sigma_g| \mid 2^i - 1\}$ or $\ell_g \leftarrow 0]$
        If $\ell_g > 0$
                If $g^{2^{\ell_g}} \neq g$
                    [In this case, $|g| = 2|\sigma_g|$. Otherwise, $\langle g \rangle = \langle \sigma_g \rangle$.]
                    Set $k_g \leftarrow$ Bounded-Order$(\sigma_g, \ell_g)$
                            $[k_g \leftarrow |\sigma_g|]$
                    If $(\ell_g = 6$ and $k_g = 63)$
                                    Or $k_g$ contains a primitive prime divisor of $2^{\ell_g} - 1$
                            If $\ell_g > \ell$
                                    Set $(\ell, t, \sigma) \leftarrow (\ell_g, g^{k_g}, \sigma_g)$
Return$(\ell, t, \sigma)$
```

The value of ℓ is a lower bound on $n-2$ because it arises through the existence of an element whose order divides $|GL(\ell, 2)|$ but not $|GL(i, 2)|$ for $1 \leq i < \ell$. If $\mathcal{S}$ does contain an element of $\mathcal{C}_2$, then `Find-Element-Of-`$\mathcal{C}_2$`-In-Set` will find it.

We are now able to describe the procedure `Find-Element-in-`$\mathcal{C}_2$. The input to `Find-Element-in-`$\mathcal{C}_2$ is a generating set $\mathcal{G}$ of G and an upper bound M on the dimension n of G. A single call is made to `Find-Element-Of-`$\mathcal{C}_2$`-In-Set` with argument a set $\mathcal{S}$ of $4M$ randomly chosen element of G. This will ensure that $\mathcal{S}$ contains an element of $\mathcal{C}_2$ with constant probability.

```
Procedure Find-Element-in-$\mathcal{C}_2$
    Input: A generating set $\mathcal{G}$ and a bound $M$ on the dimension of $G$.
    Output: A triple $(\ell, t, \sigma)$, where $\ell = n - 2$, $t$ is a transvection and $s$ commutes
        with $t$ such that $ts \in \mathcal{C}_2$.
    Reliability: $1 - 1/e$.
    Complexity: $O(M \log^2 M\mu + M\rho)$, where $\mu, \rho$ are as defined in Theorem 1.1.

Let $\mathcal{S}$ be a set of $4M$ randomly chosen elements of $G$
Set $(\ell, t, \sigma) \leftarrow$ Find-Element-in-$\mathcal{C}_2(\mathcal{S}, M)$
```

```
If $\ell \neq 0$
Return($\ell, t, \sigma$)
```

5. Preliminary Results on Transvections

There is a transvection $\tau_{v,f}$ of $GL(n,2)$ for any non-zero vector $v \in V$ and linear form $f \in V^*$, such that $f(v) = 0$. The action of $\tau_{v,f}$ on V is given by the formula

$$w\tau_{v,f} = w + f(w)v.$$

When the field is not $GF(2)$ there is a scale question. Replacing v by λv and f by $\lambda^{-1} f$ does not change $\tau_{v,f}$.

Lemma 5.1. *Two transvections $\tau_{v,f}$ and $\tau_{u,g}$:*

(i) *are the same,*
(ii) *commute and their product is $\tau_{u+v,f}$ when $f = g$, $u \neq v$,*
(iii) *commute and their product, $\tau_{v,f+g}$ when $f \neq g$, $u = v$,*
(iv) *commute and their product is not a transvection, when $f(u) = g(v) = 0$, $f \neq g$, $u \neq v$,*
(v) *have product of order 3 when $f(u) = g(v) = 1$; note that here, $\tau_{v,f}^{\tau_{u,g}} = \tau_{u+v,f+g}$,*
(vi) *have product of order 4 with square $\tau_{v,g}$ when $f(u) = 1$ and $g(v) = 0$, or*
(vii) *have product of order 4 with square $\tau_{u,f}$ when $f(u) = 0$ and $g(v) = 1$.*

Given a fixed transvection $\tau_{v,f}$, we are interested in finding a transvection $\tau_{u,g}$ such that $\tau_{v,f}\tau_{u,g}$ has a specified property.

Lemma 5.2. *Let $\tau_{u,g}$ be randomly chosen. Then for sufficiently large n, each of the following events has probability approaching 1/4 for large n.*

(i) *$\tau_{v,f}$ commutes with $\tau_{u,g}$ but $\tau_{v,f}\tau_{u,g}$ is not a transvection.*
(ii) *$|\tau_{v,f}\tau_{u,g}| = 3$.*
(iii) *$|\tau_{v,f}\tau_{u,g}| = 4$, with $(\tau_{v,f}\tau_{u,g})^2 = \tau_{v,g}$.*
(iv) *$|\tau_{v,f}\tau_{u,g}| = 4$, with $(\tau_{v,f}\tau_{u,g})^2 = \tau_{u,f}$.*

Proof. The structure of $C_{GL(n,2)}(\tau_{v,f})$ is a split extension of an extra-special 2-group of order 2^{2n-3} by $GL(n-2,2)$. Thus the number of transvections is $[GL(n,2) : C_{GL(n,2)}(\tau_{v,f})] = (2^n - 1)(2^{n-1} - 1)$. Given a fixed transvection $\tau_{v,f}$, it follows from Lemma 5.1(iv), that the number of transvections $\tau_{u,g}$ which satisfy (i) is $(2^{n-1} - 2)(2^{n-2} - 2)$. Hence, the proportion of transvections which have this property is then $(2^{n-1} - 2)(2^{n-2} - 2)/(2^n - 1)(2^{n-1} - 1)$. This approaches 1/4 asymptotically and establishes (i). The proofs of (ii), (iii) and (iv) follow in a similar manner. □

Remark 5.3. *It follows from Lemma 5.2(iii),(iv), that for a given transvection $\tau_{v,f}$ with probability 1/2, a randomly chosen transvection $\tau_{u,g}$ has the property that $\tau = (\tau_{v,f}\tau_{u,g})^2$ is a transvection. In fact, $\tau = \tau_{v,g}$ or $\tau_{u,f}$ and so both $\tau\tau_{v,f}$ and $\tau\tau_{u,g}$ are both transvections. Note that it would be difficult to find a transvection such as τ directly.*

Remark 5.4. *In the course of constructing Θ, we will often use Lemma 5.1 in the following way. Suppose that t is a transvection of G and $\Theta(t) = \tau_{v,f}$ so that t is labeled by $t = t_{v,f}$. Let t' be an arbitrary transvection of G which we label by*

$t' = t_{u,g}$ to indicate that $\Theta(t) = \tau_{u,g}$, and suppose there is no a priori relationship between v, f and u, g. If $|t_{v,f}t_{u,g}| = 4$, then $\Theta((t_{v,f}t_{u,g})^2) = (\tau_{v,f}\tau_{u,g})^2$, and so by Lemma 5.1(v),(vi), either $(\tau_{v,f}\tau_{u,g})^2 = \tau_{u,f}$ or $(\tau_{v,f}\tau_{u,g})^2 = \tau_{v,g}$. Hence, the same holds in G, namely $(t_{v,f}t_{u,g})^2 = t_{u,f}$ or $(t_{v,f}t_{u,g})^2 = t_{v,g}$.

6. Building a Dual Block Pair

In this section, we construct the dual block pair (B_Γ, B_β) required for the construction of the isomorphism Θ. The input to `Construct-Block-Basis` is the dimension n of G, a transvection t of G, which we denote by $t_{v,f}$ to indicate that $\Theta(t_{v,f}) = \tau_{v,f} \in GL(n, 2)$, an element σ which commutes with $t_{v,f}$ such that $\Theta(\sigma)$ is a *ppd*$(n, 2, n-2)$ element of $GL(n, 2)$, and a generating set $\mathcal{G}$ for G.

This construction is Las Vegas, in that if the inputs are correct (if $G \simeq GL(n, 2)$, t is a transvection, etc.), then either the answer `false` is returned or else the correct answer is returned. Thus, one does not need to verify correctness of the answer. Further, there is some constant such that the correct answer is returned with at least that probability. Thus, repeated application of the procedure can assure that if the inputs are correct, then the probability of returning `false` on all iterations can be made arbitrarily small.

There are two maximal block systems of $GL(n, 2)$ in the conjugation action on the conjugacy class of transvections. The two blocks which contain $\tau_{v,f}$ are $\Gamma(f) = \{\tau_{x,f} : x \in V, f(x) = 0, v \neq 0\}$ and $\beta(v) = \{\tau_{v,h} : x \in V, h(v) = 0, h \neq 0\}$. It is clear from Lemma 5.1 that $\Gamma(f) \cup \beta(v)$ contains all transvections which commute with $\tau_{v,f}$ and whose product with $\tau_{v,f}$ is also a transvection. In our construction, $\Theta(B_\Gamma)$ is a basis for $\langle\Gamma(f)\rangle$. Note that $\Gamma(f)$ and $\beta(v)$ are conjugate in $Aut(GL(n, 2))$, so the choice that our block maps onto $\Gamma(f)$ is purely arbitrary.

The first step in applying these ideas to the black box group G is to find another transvection in the same block as $t_{v,f}$. By Lemma 5.2, the probability is approximately $1/2$ that a random transvection $t_{u,g}$ satisfies $|t_{v,f}t_{u,g}| = 4$. But then $(t_{v,f}t_{u,g})^2 = t_{v,g}$ or $t_{u,f}$ as in Remark 5.4. Without loss of generality, we may assume that $(t_{v,f}t_{u,g})^2 = t_{u,f}$, for some $u \neq v$, and hence $t_{v,f}$ and $t_{u,f}$ belong to $\langle B_\Gamma\rangle$.

Since σ centralizes $t_{v,f}$, σ normalizes $\langle B_\Gamma\rangle$. From the structure of $C_G(t_{v,f}) \cong 2^{2n-3}GL(n-2, 2)$, σ acts irreducibly on a hyperplane of $\langle B_\Gamma\rangle$. In particular, $\langle B_\Gamma\rangle = \langle t_{v,f}\rangle \oplus [\langle B_\Gamma\rangle, \langle\sigma\rangle]$ where $\langle t_{v,f}\rangle$ and $[\langle B_\Gamma\rangle, \langle\sigma\rangle]$ are, of necessity, the only proper σ invariant subspaces of $\langle B_\Gamma\rangle$. Since σ does not centralize $t_{u,f}$, $t_{v_2,f} = [t_{u,f}, \sigma] \in [\langle B_\Gamma\rangle, \langle\sigma\rangle]$. Hence $t_{v_2,f}, t_{v_2,f}^{\sigma}, \dots, t_{v_2,f}^{\sigma^{n-2}}$ is a basis for $[\langle\Gamma(f)\rangle, \langle\sigma\rangle]$ and adding $t_{v,f}$ yields a basis for $\langle B_\Gamma\rangle$. This leads to:

`Procedure Construct-Block-Basis`

Input: $(n, t, \sigma, \mathcal{G})$ where n is the dimension of G, $t = t_{v,f}$ is a transvection of G, σ is an element which commutes with t such that σ is a *ppd*$(n, 2, n-2)$ element, and $\mathcal{G}$ is a generating set for G.

Output: A basis $B_\Gamma = \{t_{v_1,f}, \dots, t_{v_{n-1},f}\}$ for $\langle B_\Gamma\rangle$

Complexity: $O(n\mu + \rho)$, where μ, ρ are as defined in Theorem 1.1.

Reliability: $1/2$

```
Set $t_{v_1,f} \leftarrow t_{v,f}$
Set $x \leftarrow Rand(\mathcal{G})$ and $t_{u,g} \leftarrow t_{v,f}^x$
If $|t_{u,g}t_{v,f}| \neq 4$
```

```
        Return (false)
Set $t_{u,f} \leftarrow (t_{v,f}t_{u,g})^2$
Set $t_{v_2,f} \leftarrow [t_{u,f}, \sigma]$
For $i \leftarrow 3$ to $n-1$
        Set $t_{v_i,f} \leftarrow t^{\sigma}_{v_{i-1},f}$
Set $B_\Gamma \leftarrow \{t_{v_1,f}, \dots, t_{v_{n-1},f}\}$
Return($B_\Gamma$)
```

We now focus on finding a suitable vector v_n such that $\{v_1, \dots, v_n\}$ is a basis for V and then computing a set B_β such that $\Theta(B_\beta)$ is basis for $\langle\beta(v_n)\rangle$. We begin by observing from Lemma 5.1 that the probability is 1/4 that a randomly chosen transvection $t_{v_n,g'_{n-1}}$ satisfies $|t_{v,f}t_{v_n,g'_{n-1}}| = 3$, and for such a transvection, $f(v_n) = 1$. Assuming that $t_{v_n,g'_{n-1}}$ satisfies this property, it follows that $v_1, \dots, v_{n-1}$ together with v_n forms a basis for V. We will first construct a basis $B'_\beta = \{t_{v_n,g'_1}, \dots, t_{v_n,g'_{n-1}}\}$ such that $\Theta(B'_\beta)$ is a basis for $\beta(v_n)$ and then transform this to a basis $B_\beta = \{t_{v_n,g_1}, \dots, t_{v_n,g_{n-1}}\}$ with the desired property that $g_i(v_j) = \delta_{ij}$, $1 \le i, j \le n-1$.

If $t_{v_n,g'_{n-1}} = t^x_{v,f}$, then conjugating B_Γ by x will give a basis for $\Theta^{-1}(\langle\Gamma(g'_{n-1})\rangle)$. A randomly chosen transvection $t_{u,g'}$ has probability 1/2 of satisfying the equation $|t_{u,g'}t_{v_n,g'_{n-1}}| = 4$. In this case, either (i) $g'_{n-1}(u) = 0, g'(v_n) = 1$ or (ii) $g'_{n-1}(u) = 1, g'(v_n) = 0$. In case (i), $|t_{u,g'}t'| \mid 4$ for all $t_{w,g'_{n-1}} \in \Theta^{-1}(\langle\Gamma(g'_{n-1})\rangle)$. Note that $t_{u,g'}$ has equal likelihood of being in case (i) or case (ii). Assuming we can confirm that $t_{u,g'}$ is in case (ii), it then follows that $(t_{u,g'}t_{v_n,g'_{n-1}})^2 = t_{v_n,g'} \in \Theta^{-1}(\beta(v_n))$. We may then construct the basis $B'_\beta = \{t_{v_n,g'_1}, \dots, t_{v_n,g'_{n-1}}\}$ for $\Theta^{-1}(\beta(v_n))$ by setting $t_{v_n,g'_1} = t_{v_n,g'}$ and $t_{v_n,g'_{i+1}} = t_{v_n,g'_i}{}^{\sigma^x}$, $1 \le i \le n-2$.

It remains for us to verify that $t_{u,g'}$ is in case (ii). We first observe that case (ii) can be distinguished from case (i) since $|t_{u,g'}t_{w',g'_{n-1}}| = 3$ for approximately 1/2 of the elements $t_{w',g'_{n-1}} \in \Theta^{-1}(\Gamma(g'_{n-1}))$. To see that this happens in case (ii), note that $g'_{n-1}(u) = 1$ implies that $|t_{u,g'}t_{w,g'_{n-1}}| = 4$ if $g'(w) = 0$ and $|t_{u,g'}t_{w,g'_{n-1}}| = 3$ if $g'(w) = 1$. Since, $g'(v_n) = 1$ and $v_n \in ker(g_{n-1})$, g' is not trivial on $ker(g'_{n-1})$. Hence, $g'(w) = 1$ on 1/2 of the vectors $w \in ker(g'_{n-1})$ which proves the assertion.

Thus our test for finding a transvection $t_{u,g'}$ in case (ii) consists of the following. First generate a random transvection $t_{u,g'}$ and test if $|t_{u,g'}t_{v_n,g_{n-1}}| = 4$. If so, then generate a random element $t_{w,g'_{n-1}} \in \Theta^{-1}(\Gamma(g'_{n-1}))$ and check if $|t_{w,g'_{n-1}}t_{u,g'}| = 3$. Thus the overall probability of generating a random transvection which can then be confirmed to be in case (ii) is at least 1/8.

We now describe how to transform B'_β to B_β. We want to find an $(n-1)\times(n-1)$ matrix $X = (x_{ij})$ such that if

$$t_{v_n,g_i} = \prod_{j=1}^{n-1} t^{x_{ij}}_{v_n,g'_j}$$

then $g_i(v_j) = \delta_{ij}$, $1 \le i, j \le n-1$. Note that this is equivalent to finding (x_{ij}) such that

$$g_i = \sum_{j=1}^{n-1} x_{ij}g'_j$$

subject to $g_i(v_j) = \delta_{ij}$, $1 \le i,j \le n-1$. Let $Y = (y_{ij})$ be the $(n-1) \times (n-1)$ matrix with $y_{ij} = g'_i(v_j)$, $1 \le i,j \le n-1$. It is easy to see that Y is non-singular since $\{g'_1, \dots, g'_{n-1}\}$ is a basis for the annihilator of $\langle v_n \rangle$ in V^* and hence remains independent when viewed as a subset of $\langle v_1, \dots, v_{n-1} \rangle^*$. It then follows by direct computation that $X = Y^{-1}$. This reduces the problem of determining X to computing Y. Now although we don't have an explicit representation of the functionals $\{g'_1, \dots, g'_{n-1}\}$ or the vectors $\{v_1, \dots, v_{n-1}\}$, we can still compute $y_{ij} = g'_i(v_j)$. Since $f(v_n) = 1$, it follows from Lemma 5.1 that $g'_i(v_j) = 0$ if $|t_{v_n,g'_i} t_{v_j,f}| = 4$ and $g'_i(v_j) = 1$ if $|t_{v_n,g'_i} t_{v_j,f}| = 3$.

`Procedure Construct-Dual-Block-Pair`

Input: $(n, t, \sigma, B_\Gamma, \mathcal{G})$ where n, t, σ, $\mathcal{G}$ are as before and B_Γ is the basis for $\Theta^{-1}(\langle \Gamma(f) \rangle)$ constructed in `Construct-Block-Basis`.
Output: A basis $B_\beta = \{t_{v_n,g_1}, \dots, t_{v_n,g_{n-1}}\}$ for $\Theta^{-1}(\langle \beta(v_n) \rangle)$ where $f(v_n) = 1$ and $g_i(v_j) = \delta_{ij}$, $1 \le i,j \le n-1$.
Complexity: $O(n^2\mu + \rho + n^3\epsilon)$, where μ, ρ, ϵ are as defined in Theorem 1.1.
Reliability: 1/32

```
Set  $x \leftarrow Rand(\mathcal{G})$
Set $t_{v_n,g'_{n-1}} \leftarrow t^x_{v,f}$
If $|t_{v_n,g'_{n-1}} t_{v,f}| \ne 3$
    Return(false)
    [Thus $f(v_n) = 1$, $g'_{n-1} \ne f$, and $\{v_1, \dots, v_n\}$ is a basis for $V$.]
Set $t_{w,g'_{n-1}} \leftarrow {t_{v_2,f}}^x$
    [$v_2 \ne v$ implies that $w \ne u$]
Set $x' \leftarrow Rand(\mathcal{G})$ and $t_{u,g'} \leftarrow t^{x'}_{v,f}$
If ($|t_{u,g'} t_{v_n,g'_{n-1}}| \ne 4$ or $|t_{w,g'_{n-1}} t_{u,g'}| \ne 3$)
    Return(false)
    [This ensures that $t_{u,g'}$ satisfies case (ii) in the previous discussion]
Set $t_{v_n,g'_1} \leftarrow (t_{u,g'} t_{v_n,g'_{n-1}})^2$
    [$t_{v_n,g'_1}$ is an additional element of $\Theta^{-1}(\beta(v_n))$]
For $i \leftarrow 1$ to $n-2$
    Set $t_{v_n,g'_{i+1}} \leftarrow {t_{v_n,g'_i}}^{\sigma^x}$
Set $B'_\beta \leftarrow \{t_{v_n,g'_1}, \dots, t_{v_n,g'_{n-1}}\}$
    [Construct $B_\beta$]
For $i \leftarrow 1$ to $n-1$
    For $j \leftarrow 1$ to $n-1$
        If $|t_{v_n,g'_i} t_{v_j,f}| = 4$
            Set $y_{ij} \leftarrow 0$
        Else Set $y_{ij} \leftarrow 1$
            [In this case, $|t_{v_n,g'_i} t_{v_i,f}| = 3$]
Set $(x_{ij}) \leftarrow (y_{ij})^{-1}$
    [Viewing $(x_{ij})$ and $(y_{ij})$ as matrices]
For $i \leftarrow 1$ to $n-1$
    Set $t_{v_n,g_i} = \prod_{j=1}^{n-1} t^{x_{ij}}_{v_n,g'_j}$
Return($B_\beta$)
```

7. Computing the Center of a Transvection

We will assume the existence of the dual block pair (B_Γ, B_β) constructed in the previous section. Given an unknown transvection, $t_{w,h} \in G$, we wish to express the vector $w \in V$ as a linear combination $w = \sum_{i=1}^{n} a_i v_i$. This is accomplished by the procedure `Compute-Transvection-Center`. Similarly, we can also write h as a linear combination of the basis $\{f, g_1, \dots, g_{n-1}\}$ for V^*, but this is not required.

`Compute-Transvection-Center` has two component procedures. The first procedure, denoted by `Reduce-Transvection-Center`, has input $t_{w,h}$, B_Γ and B_β, and returns a pair $(t_{w',f}, a_n)$ where $w' \in \langle v_1, \dots, v_{n-1}\rangle$ and $w = w' + a_n v_n$. We include the possibility that $w' = 0$ in which case $t_{w',f} = 1$. The second procedure `Sift` has as input $t_{w',f}$, B_Γ and B_β, and returns coefficients $(a_1, \dots, a_{n-1})$ such that $w' = \sum_{i=1}^{n-1} a_i v_i$.

```
Procedure Compute-Transvection-Center
    Input: An unknown transvection $t_{w,h}$, $B_\Gamma$ and $B_\beta$.
    Output: Coefficients $(a_1, \dots, a_n)$ such that $w = \sum_{i=1}^{n} a_i v_i$.
    Complexity: $O(n\mu)$, where $\mu$ is as defined in Theorem 1.1.

Set $(t_{w',f}, a_n) \leftarrow$ Reduce-Transvection-Center($t_{w,h}, B_\Gamma, B_\beta$)
If $t_{w',f} = 1$ Return $(0, \dots, 0, a_n)$
Else Set $(a_1, \dots, a_{n-1}) \leftarrow$ Sift($t_{w',f}, B_\Gamma, B_\beta$)
Return $(a_1, \dots, a_n)$
```

7.1. The Procedure `Reduce-Transvection-Center`.

We will first present a conceptual development of the procedure, including a proof of correctness and then present the pseudocode. Starting with an arbitrary transvection $t_{w,h}$, the goal is to find a transvection $t_{w',f}$ where $w' = w + a_n v_n \in \langle v_1, \dots, v_{n-1}\rangle$. This is accomplished through the following sequence of cases.

Case 1 $w = v_n$, hence $w' = 0$.

This case can be determined by testing if $|t_{w,h} t_{v_n,g_i}| = 2$ for all i, $1 \leq n-1$. If this happens, then $g_i(v) = 0$, $1 \leq i \leq n-1$, in which case $\beta(w) = \beta(v_n)$. Hence $w = v_n$. The procedure will then return $(t_{0,h}, 1)$.

Case 2 $f(w) = 0$.

We first show that $f(w) = 0$ if and only if $|t_{w,h} t_{v_i,f}| \mid 4$ for all i, $1 \leq i \leq n-1$. This provides a simple test for this case. First note that this condition is true if and only if $f(w) = 0$. To see this, note that by Lemma 5.1, $f(w) = 0$ implies $|t_{w,h}, t_{v_i,f}| \in \{2, 4\}$ for all i. Conversely, assume that $|t_{w,h} t_{v_i,f}| \in \{2, 4\}$ for all i and $f(w) = 1$. Then $h(v_i) = 0$ for all i by Lemma 5.1 and so $h \in \bigcap_{i=1}^{n-1} v_i^\circ = \langle f\rangle$. But then $h = f$ contradicting $h(w) = 0$. In this case, $a_n = 0$ and so $w' = w$. Further reduction of $t_{w',h}$ to $t_{w',f}$ is performed in case 4.

Case 3 $f(w) = 1$.

Since $f(v_n) = 1$ by our choice of v_n, and $f(w) = 1$ by assumption, w involves v_n and $a_n = 1$. Suppose first that $|t_{w,h}, t_{v_n,g_i}| = 3$ for some i, $1 \leq i \leq n-1$. Then $t_{w',h'}$ is set to $t_{w,h}^{t_{v_n,g_i}} = t_{w+v_n,h+g_i}$. In the remaining case, $|t_{w,h}, t_{v_n,g_i}| \mid 4$, $1 \leq i \leq n-1$. We claim that $h(v_n) = 0$. Otherwise, $h(v_n) = 1$ which implies that $g_i(w) = 0$, $1 \leq i \leq n-1$. But then, $w = v_n$, which contradicts $h(w) = 0$. We may further assume that since $f(w) = 1$, $g_i(w) = 1$ for some i, $1 \leq i \leq n-1$ and hence $|t_{w,h}, t_{v_n,g_i}| = 4$. In this case, $t_{w',h'}$ is set to $t_{w,h}(t_{w,h}t_{v_n,g_i})^2 = t_{w+v_n,h}$. This reduces to the case where $f(w') = 0$ and is addressed next.

Case 4 $f(w') = 0$.

We have reduced to the case of analyzing $t_{w',h'}$ where $f(w') = 0$. Note that the case $w' = 0$ has already been discussed in case 1. We know at this point that $w' = w + a_n v_n$ for our original transvection $t_{w,h}$.

Recall from case 2 that $|t_{w',h'}t_{v_i,f}| \mid 4$ for all i, $1 \leq i \leq n-1$. If there is an i such that $|t_{w',h'}t_{v_i,f}| = 4$, then $(t_{w',h'}t_{v_i,f})^2 = t_{w',f}$ and we return $((t_{w',h'}t_{v_i,f})^2, a_n)$. Otherwise, $|t_{w',h'}t_{v_i,f}| = 2$ for all i, $1 \leq i \leq n-1$ which implies that $h' = f$ and we return $(t_{w',h'}, a_n)$.

`Procedure Reduce-Transvection-Center`

Input: An unknown transvection $t_{w,h}$, B_Γ and B_β.

Output: $(t_{w',f}, a_n)$ where $a_n = f(w)$ and $w = w' + a_n v_n$. (Recall that B_Γ uniquely determines f.) [Note that we also allow $w' = 0$, in which case $t_{w',f} = 1$.]

Complexity: $O(n^2\mu)$, where μ is as defined in Theorem 1.1.

```
If $|t_{w,h}t_{v_n,g_i}| = 2$ for all $i$, $1 \leq n-1$ [$w = v_n$]
    Return $(t_{0,h}, 1)$
If $|t_{w,h}t_{v_i,f}| \mid 4$ for all $i$, $1 \leq i \leq n-1$ [$f(w) = 0$]
    Set $t_{w',h'} \leftarrow t_{w,h}$
    Set $a_n \leftarrow 0$
Else [$f(w) = 1$]
    Set $a_n \leftarrow 1$
    If $|t_{w,h}t_{v_n,g_i}| = 3$ for some $i$, $1 \leq n-1$ [$h(v_n) = 1$]
        Set $t_{w',h'} \leftarrow t_{w,h}^{t_{v_n,g_j}}$ [$w' = w + v_n$]
    Else Choose $j$, $1 \leq j \leq n-1$ such that $|t_{w,h}t_{v_n,g_j}| = 4$
        Set $t_{v_n,h} = (t_{w,h}t_{v_n,g_j})^2$
        Set $t_{w',h'} \leftarrow t_{w,h}t_{v_n,h}$
For $i \leftarrow 1$ to $n-1$ [Now, $f(w') = 0$]
    If $|t_{w',h'}t_{v_i,f}| = 4$
        Return $((t_{w',h'}t_{v_i,f})^2, a_n)$ [$(t_{w',h'}t_{v_i,f})^2 = t_{w',f}$]
Return $(t_{w',h'}, a_n)$ [$h'(v_i) = 0$ for all $i$, hence $h' = f$.]
```

7.2. The Procedure `Sift`. The key to the procedure `Sift` is the choice of basis $B_\beta = \{t_{v_n,g_1}, \ldots, t_{v_n,g_{n-1}}\}$ for $\Theta^{-1}(\langle\beta(v_n)\rangle)$ with the property that $g_i(v_j) = \delta_{ij}$, $1 \leq i, j \leq n-1$. In this case, if $w = \sum_{j=1}^{n-1} a_j v_j$, then $g_i(w) = a_i$. Since $f(v_n) = 1$, $|t_{w,f}t_{v_n,g_i}| = 3$ or 4 depending on whether $a_i = g_i(w) = 1$ or 0 respectively.

`Procedure Sift`

Input: A transvection $t_{w,f} \in \Theta^{-1}(\Gamma(f))$, B_Γ and B_β.

Output: A sequence $(a_1, \ldots, a_{n-1})$ with each $a_i \in GF(2)$ such that $w = \sum_{i=1}^{n-1} a_i v_i$ (or equivalently, such that $t_{w,f} = t_{v_1,f}^{a_1} \cdots t_{v_{n-1},f}^{a_{n-1}}$).

Complexity: $O(n\mu)$, where μ is as defined in Theorem 1.1.

```
For $i \leftarrow 1$ to $n-1$
    If $|t_{w,f}t_{v_n,g_i}| = 3$
```

```
                Set $a_i \leftarrow 1$
          Else Set $a_i \leftarrow 0$
                [In this case $|t_{w,f}t_{v_n,g_i}| = 4$]
Return($(a_1, \ldots , a_{n-1})$)
```

8. Open Questions

We enumerate questions which emerge from this work and which will be the subject of future research efforts on our part.

Question 8.1. *The most interesting question concerns the extension of Theorem 1.1 to groups defined over prime fields for odd primes p. There are two possible such extensions.*

- *Construct the natural representation into $GL(n, p)$ for a black box group G satisfying $SL(n, p) \subseteq G \subseteq GL(n, p)$.*
- *Construct the natural projective representation into $GL(n, p)$ for a black box group G satisfying $PSL(n, p) \subseteq G \subseteq PGL(n, p)$ (for the case when $SL(n, p)$ is not simple).*

It appears that the machinery set up in this paper should settle the two issues raised in Question 8.1, at least for the special linear groups. Such an extension is almost complete.

Question 8.2. *Does Theorem 1.1 extend to the other classical groups?*

Question 8.3. *Can the algorithm used to prove Theorem 1.1 be made Las Vegas?*

As indicated in the introduction, proving the correctness of Θ relies solely on verifying the value of n determined by the algorithm. We sketch a two-part approach that verifies n together with the underlying hypothesis that $G \simeq GL(n, 2)$.

(1): Construct a subgroup G_0 of G with $G_0 \simeq GL(n, 2)$.
(2): Prove that $G = G_0$

A standard approach to solving (1) is to first find a presentation $\mathcal{P} = \langle \mathcal{S}|\mathcal{R}\rangle$ for $GL(n, 2)$ and a set S of elements satisfying the presentation $\mathcal{P}$, i.e. there exists a 1-1 map $\phi : \mathcal{S} \longrightarrow S$ so that each relation of $\mathcal{R}$ is satisfied in G if elements of $\mathcal{S}$ in the relation are replaced by their respective images under ϕ. This approach works as long as $n > 2$, since $GL(n, 2)$ is then simple. Given the machinery that has already been developed, it should be possible to construct a set S of transvections in G satisfying $\mathcal{P}$.

In order to answer (2), it suffices to show that each generator $g \in \mathcal{G}$ of G is an element of G_0. A recent result of Celler and Leedham-Green [**6**] seems to apply directly to this problem. We first compute $\Theta(g)$ and then apply [**6**] to express $\Theta(g)$ as word in $\Theta(S)$. By shadowing the computation in G, we can then produce an element $g_0 \in G_0$ such that $\Theta(g_0) = \Theta(g)$. We then test if $g_0 = g$ to certify that $g \in G_0$.

References

[1] L. Babai, "Local expansion of vertex-transitive graphs and random generation in finite groups", *Proc. 23rd ACM STOC*, pp. 164–174.

[2] L. Babai, and R. Beals, (1993). "Las Vegas Algorithms for Matrix Groups,", *Proc. 24th IEEE FOCS*, pp. 427–436.

[3] R. Beals, "Towards polynomial time algorithms for matrix groups", in: *Groups and Computation II* (L. Finkelstein, W. M. Kantor, eds.), DIMACS Workshop on Groups and Computation (June 1995), A. M. S., to appear.

[4] F. Celler, C. R. Leedham-Green, S .H. Murray, A C. Niemeyer, and E. A. O'Brien, " Generating random elements of a matrix group", Comm. Algebra, **23**, 1995, 4931–4948.

[5] F. Celler and C. R. Leedham-Green, "Calculating the Order of an Invertible Matrix", in: *Groups and Computation II* (L. Finkelstein, W. M. Kantor, eds.), DIMACS Workshop on Groups and Computation (June 1995), A. M. S., to appear.

[6] F. Celler and C. R. Leedham-Green, "A Constructive Recognition Algorithm for the Special Linear Group", preprint.

[7] F. Celler and C. R. Leedham-Green, "A Non-Constructive Recognition Algorithm for the Special Linear and Other Classical Groups", in: *Groups and Computation II* (L. Finkelstein, W. M. Kantor, eds.), DIMACS Workshop on Groups and Computation (June 1995), A. M. S., to appear.

[8] P.M. Neumann and C. Praeger, "A recognition algorithm for special linear groups", *Proc. London Math, Soc.* **65** (1992), 555-603.

[9] A.C. Niemeyer and Cheryl Praeger, "Implementing a Recognition Algorithm for Classical Groups", in: *Groups and Computation II* (L. Finkelstein, W. M. Kantor, eds.), DIMACS Workshop on Groups and Computation (June 1995), A. M. S., to appear.

College of Computer Science, Northeastern University, Boston, MA 02115,
E-mail address: `gene@ccs.neu.edu`

College of Computer Science, Northeastern University, Boston, MA 02115,
E-mail address: `laf@ccs.neu.edu`

Department of Computer Science, St. Andrews University, St. Andrews, Scotland,
E-mail address: `sal@cs.st-andrews.ac.uk`