

A Constructive Recognition Algorithm for the Matrix Group

$$\Omega(d, q)$$

Peter A. Brooksbank *

June 6, 2004

Abstract

We present an efficient algorithm to recognise constructively the orthogonal groups $\Omega(d, q)$ in their natural representation.

1 Introduction

The matrix group recognition project is a broad and complex area of research, incorporating the work of many different authors (cf. [LG]). Among the key ingredients is the need to recognise when a matrix group G , given by arbitrary generators, is a classical group, and, if it is, to use this fact to compute efficiently inside G using the given generators. This latter requirement leads to the notion of a *constructive* recognition algorithm. Namely, suppose that the matrix group $G = \langle \mathcal{S} \rangle \leq \text{GL}(d, q)$ is a classical group. The objective, for a given element g of G , is to write g as a word in \mathcal{S} . However, such a word may be very long, so instead we will find a *straightline program* from \mathcal{S} to g .

A straightline program, or *SLP*, from \mathcal{S} is a sequence of elements of $G = \langle \mathcal{S} \rangle$, where each element appearing in the sequence is either in \mathcal{S} , or is the inverse of an earlier element in the sequence, or else is the product of two earlier elements. If $g \in G$ is the last element of the SLP, then we say that g has been *constructed using an SLP from \mathcal{S}* ; we will call the sequence an *SLP from \mathcal{S} to g* .

Celler and Leedham-Green [CeLG] provided the first example of a constructive recognition algorithm when G is known to contain $\text{SL}(d, q)$, and Celler [Ce] later dealt with $G = \text{Sp}(d, q)$. Both algorithms have been implemented, and are available in GAP3.

This type of problem arises in various algorithmic settings (for example, for quotients of matrix groups and for finding Sylow subgroups in permutation groups [Ka, KLM, Ma, Mo]). In

*This research was supported in part by the National Science Foundation.

an attempt to solve many such problems at once, constructive recognition has been addressed in the much more general framework of *black box* groups. In [Bt, BrK, CFL] constructive recognition algorithms are given for black box groups isomorphic to perfect central extensions of $\mathrm{PSL}(d, q)$, while [KS] gives algorithms for all black box classical groups.

In this note, we remain within the context of matrix groups, and outline an algorithm for constructive recognition of all classical groups in their natural representation. In this setting, we are able to use linear algebra more efficiently than is possible inside a black box group, and this gives rise to faster algorithms (cf. Remark (i) below). Furthermore, with a little work, these matrix group algorithms can be inserted into versions of black box algorithms in [KS], obviating the need for a recursive call.

Although this paper only deals with the groups $\Omega(d, q)$, the algorithm is easily modified to deal with all classical groups. Moreover, when $G = \mathrm{Sp}(d, q)$ our algorithm has improved asymptotic running time over the algorithm in [Ce]. The author has implemented this algorithm for the orthogonal groups in GAP4, and the implementation of the remaining classical groups will be completed soon. We will discuss the main modifications required for the latter in section 5.

Our algorithm employs both Lie theory and linear algebra and we often change our perspective in order to obtain a clearer understanding of different segments of the algorithm. Moreover, we bring together ideas from several algorithmic sources. For example, the central technique of the algorithm can be viewed as a synthesis of the point-stabiliser methods in [KS] and the m -space stabiliser methods in [Ce].

Primitive prime divisor methods akin to those in [KS] are employed to remove all explicit occurrences of q in the running time. In particular, we show that the polynomial time bottleneck is now the efficient treatment of the low dimensional cases. The recent advance of Conder and Leedham-Green [CoLG], reported at this conference, emphasises the practical importance of this seemingly theoretical observation. They present a fast recognition algorithm for $\mathrm{SL}(2, q)$ as a matrix group in its natural representation using an oracle for solving discrete logarithm problems in $\mathrm{GF}(q)^*$. Here, we assume the availability of an oracle for constructive recognition of $\mathrm{SL}(2, q)$, implicitly assuming that this means a discrete logarithm oracle is also available.

Our main result can be stated as follows.

Theorem 1.1 *Let $G = \langle \mathcal{S} \rangle \leq \mathrm{GL}(d, q) = \mathrm{GL}(V)$, where $q = p^k$ for a prime p and a positive integer k , and G is known to be $\Omega(d, q)$, preserving some (unknown) quadratic form on V . In Las Vegas time*

$$O(d^3 \log d \log^4 q + \xi d + \mu d^2 \log q + \chi d \log q),$$

the elements of a new generating set \mathcal{T} , of size $O(kd^2)$, can be constructed using straightline programs (SLPs) from \mathcal{S} , such that there is a deterministic algorithm to write an SLP of length $O(d^2 \log q)$ from \mathcal{T} to any given element g of G in time $O(d^3 \log q + \chi)$.

Here, ξ is an upper bound on the time requirement per element for the construction of independent, (nearly) uniformly distributed random elements of G , μ is an upper bound on the time required to perform each group operation in G , and χ represents the cost, per call, to $\text{SL}(2, q)$ and discrete logarithm oracles.

Remarks. (i) The timing in [KS] for the black box recognition of the groups $\text{P}\Omega^\epsilon(d, q)$, even without verifying a presentation, is $O(\xi q d \log d(d + \log q) + \mu q d^4 \log^2 q \log^3 d)$. [KS], **3.6.1** gives a black box algorithm for constructive recognition of $\text{SL}(2, q)$ in time $\chi = O(\xi q \log q + \mu q \log^2 q)$. Even using this estimate of χ and estimating discrete logs as $O(q)$, the complexity of our algorithm is $O(d^3 \log d \log^4 q + \xi(d + q \log q) + \mu(d^2 \log q + q \log^2 q))$: our algorithm for $\Omega^\epsilon(d, q)$ in its natural representation runs significantly faster.

(ii) The basic idea of the algorithm, as in [CeLG, Ce], is to construct a set \mathcal{T} from which it is easy to write an SLP, of modest length, to any given element of G . In [CeLG], \mathcal{T} consists of sufficiently many transvections to perform Gaussian elimination in $\text{SL}(d, q)$. *We will describe a more involved analogue of Gaussian elimination for $\Omega(d, q)$, and the set \mathcal{T} will consist of the elements necessary to execute this procedure.*

(iii) We use the parameter μ in order to keep a count of the number of group operations in G that are required, but we note that, since $G \leq \text{GL}(d, q)$, $\mu = O(d^3 \log q)$. As in [KS], we assume that $\xi > \mu|\mathcal{S}|$.

(iv) The definition of straightline program given here implies that each element in the sequence is obtained as a matrix by computing inside G . In practice, however, it is not always necessary to write each entry in the sequence as a group element. We will often know the element (matrix) to which we are writing the SLP in advance, and simply wish to record how the element is *constructed* from a certain set of elements. In such instances, the time required to write the SLP will not involve the parameter μ .

(v) The result is stated for all groups $\Omega(d, q)$, but here we only give a proof for d odd. Of course this means that we need not consider $p = 2$, since in this case, $\Omega(d, q) \cong \text{PSp}(d - 1, q)$. Nevertheless, we still encounter most of the technical difficulties apparent in the general case while keeping the exposition as simple as possible.

(vi) Either by using Meataxe methods [HR], or alternatively by a polynomial time deterministic algorithm due to Luks, one can construct a G -module isomorphism between the natural module of G and its dual. *We therefore assume that we can find a matrix Φ corresponding to a quadratic form on V preserved by G .*

(vii) The fast Monte Carlo algorithm in [BCFLS] for computing the derived group of a matrix group G allows us to extend this result to groups G with $\Omega(d, q) \leq G \leq N_{\text{GL}(d, q)}(\Omega(d, q))$.

2 Background

We assume a basic familiarity with the classical groups and the geometries associated with them. We generally adhere to the definitions and terminology of [Ta]. Since we only prove Theorem 1.1 in the case when d and q are odd, we restrict our summary of the necessary background material to this case alone.

Let $G = \Omega(d, q)$, where d is odd, and $q = p^k$ for some odd prime p and positive integer k . Let V denote the underlying vector space of dimension d over $\mathbb{F} = \text{GF}(q)$. Let $\varphi : V \rightarrow \mathbb{F}$ denote a nonsingular quadratic form on V , preserved by G . We say that the subspace W of V is *totally singular* (t.s.) in case $\varphi(W) = 0$.

For the remainder of the paper, ρ will always denote a fixed generator of \mathbb{F}^* , and \mathbb{F}_p will denote the prime subfield of \mathbb{F} .

2.1 Standard bases

Let $m = \lfloor \frac{d}{2} \rfloor$. Replacing φ with $\lambda\varphi$ for non-square $\lambda \in \mathbb{F}$ if necessary, there exists an ordered basis for V of the form $\mathfrak{B} = \{e_1, \dots, e_m, v, f_1, \dots, f_m\}$, where $\varphi(v) = 1$ and $\varphi(e_i) = \varphi(f_i) = (e_i, e_j) = (f_i, f_j) = (e_i, v) = (f_i, v) = 0$, $(e_i, f_j) = \delta_{ij}$ for $1 \leq i, j \leq m$. Such a basis \mathfrak{B} is called a *standard basis* for V .

Fix a standard basis \mathfrak{B} for V . Relative to \mathfrak{B} , all elements of G have the form

$$g = \begin{pmatrix} A & z_1^t & B \\ z_2 & \lambda & z_3 \\ C & z_4^t & D \end{pmatrix}, \quad (1)$$

where $A, B, C, D \in \mathbb{M}_{m \times m}(\mathbb{F})$, $z_i \in \mathbb{F}^m$ for $1 \leq i \leq 4$ and $\lambda \in \mathbb{F}$. Simple matrix relations exist between the entries of g , arising from the fact that g preserves φ ; we will use some of these relations later to study the structure of certain key subgroups of G .

Let \mathfrak{B}_ρ be the \mathbb{F}_p -basis obtained from \mathfrak{B} by replacing each element w of \mathfrak{B} with the k vectors $\rho^l w$ for $0 \leq l < k$. \mathfrak{B}_ρ is ordered lexicographically, by the ordering of \mathfrak{B} first, then by increasing values of l .

2.2 Point stabilisers

Let $x = \langle e_1 \rangle$ and $y = \langle f_1 \rangle$. Then the point stabiliser

$$G_x = Q(x) \rtimes G_{x,y}, \quad (2)$$

where $Q(x) = O_p(G_x)$ is the group of isometries inducing 1 on x and on x^\perp/x . $Q(x)$ has order q^{d-2} and inherits the structure of an orthogonal space of the same isometry type as V . In

fact, $Q(x)$ is isometric to the $d - 2$ -space $\langle e_1, f_1 \rangle^\perp$ and we have the following $G_{x,y}$ -invariant correspondences:

$$\left\{ \begin{array}{l} \text{singular points} \\ \langle w \rangle \text{ in } \langle e_1, f_1 \rangle^\perp \end{array} \right\} \longleftrightarrow \left\{ \begin{array}{l} \text{'long root groups'} \\ \text{of } Q(x) \text{ associated to} \\ \text{the t.s. lines } \langle e_1, w \rangle \end{array} \right\}$$

$$\left\{ \begin{array}{l} \text{nonsingular points} \\ \langle w \rangle \text{ in } \langle e_1, f_1 \rangle^\perp \end{array} \right\} \longleftrightarrow \left\{ \begin{array}{l} \text{'root groups'} \\ \text{of } Q(x) \text{ associated to} \\ \text{the lines } \langle e_1, w \rangle \end{array} \right\} \quad (3)$$

The elements of certain root groups will play the rôle of elementary transvections in our analogue of Gaussian elimination.

Let $w \in \langle e_1, f_1 \rangle^\perp$ be singular. The long root group $R_{e_1 w}$ associated to the t.s. line $\langle e_1, w \rangle$ consists of the linear transformations

$$r_{e_1 w}(\lambda) : u \mapsto u - \lambda(u, w)e_1 + \lambda(u, e_1)w, \quad (4)$$

for $\lambda \in \mathbb{F}$. Next, let $w \in \langle e_1, f_1 \rangle^\perp$ be nonsingular. The root group $R_{e_1 w}$ associated to the line $\langle e_1, w \rangle$ consists of the linear transformations

$$r_{e_1 w}(\lambda) : u \mapsto u - \lambda(u, w)e_1 + \lambda(u, e_1)w - \lambda^2 \varphi(w)(u, e_1)e_1 \quad (5)$$

The map $\psi_x : \langle e_1, f_1 \rangle^\perp \rightarrow Q(x)$ sending $w \mapsto r_{e_1 w}(1)$ defines an isometry giving rise to the correspondences in (2.3). We similarly define an isometry $\psi_y : \langle e_1, f_1 \rangle^\perp \rightarrow Q(y)$. Starting with the standard \mathbb{F}_p -basis \mathfrak{B}_ρ (cf. 2.1), we use these isometries to obtain generating sets $B_x = \psi_x(\mathfrak{B}_\rho)$ and $B_y = \psi_y(\mathfrak{B}_\rho)$ for $Q(x)$ and $Q(y)$ respectively. We refer to the pair (B_x, B_y) as a *standard opposite pair* for $Q(x)$ and $Q(y)$.

2.3 m -space stabilisers

Let E and F be the t.s. m -spaces $E = \langle e_1, \dots, e_m \rangle$ and $F = \langle f_1, \dots, f_m \rangle$. Then the subspace stabilisers G_E and G_F are such that

$$G_E = U(E) \rtimes L \quad \text{and} \quad G_F = U(F) \rtimes L.$$

Here $U(E)$ (resp. $U(F)$) is the p -core of G_E (resp. G_F) and $L = G_{E,F}$, the subgroup of G stabilising E and F . Consider the embedding $\iota : \text{GL}(m, q) \hookrightarrow \text{O}(d, q)$ via $\iota : A \mapsto \text{diag}(A, 1, A^{-t})$. Let H denote the subgroup of index 2 of $\text{GL}(m, q)$ consisting of all matrices having square determinant. Then $L = \iota(H)$.

$U = U(F)$ consists of all matrices of the form $u(z, M) = \begin{pmatrix} I & z^t & M \\ & 1 & -2z \\ & & I \end{pmatrix}$, where $M + M^t + 2z^t z = 0$. Hence $Z(U) = \{ u(0, M) \mid M + M^t = 0 \}$ is isomorphic to the \mathbb{F} -space of all skew-symmetric matrices. $U/Z(U) \cong \mathbb{F}^m$ and $U(E) = U(F)^t$.

Generators for U: $Z(U)$ is generated by the long root groups $R_{f_i f_j}$. Observe that $F \cong U/Z(U)$ via $f \mapsto r_{f_v}(1)Z(U)$, where $r_{f_v}(1)$ is an element of the short root group R_{f_v} associated to the line $\langle f, v \rangle$. It follows that U is generated by the set

$$\mathcal{T}_F = \{ r_{f_i f_j}(\rho^l), r_{f_i v}(\rho^l) \mid 1 \leq i < j \leq m, 0 \leq l < k \}. \quad (6)$$

Similarly, $U(E)$ is generated by the set

$$\mathcal{T}_E = \{ r_{e_i e_j}(\rho^l), r_{e_i v}(\rho^l) \mid 1 \leq i < j \leq m, 0 \leq l < k \}. \quad (7)$$

Generators for L: $L = \iota(H)$ and $H/\mathrm{SL}(m, q) \cong \langle \mathrm{diag}(\rho^2, 1, \dots, 1) \rangle$. Let $a = \iota(\mathrm{diag}(\rho^2, 1, \dots, 1)) \in L$. The image, under ι , of the elementary transvection $x_{ij}(\lambda) = I + E_{ij}(\lambda)$ for $i \neq j$, is the element $r_{f_i e_j}(\lambda)$ in the long root group $R_{f_i e_j}$ associated to the t.s. line $\langle f_i, e_j \rangle$. Hence, L is generated by the set

$$\mathcal{T}_L = \{ a, r_{f_i e_j}(\rho^l) \mid 1 \leq i \neq j \leq m, 0 \leq l < k \}. \quad (8)$$

The generating set \mathcal{T} of Theorem 1.1. will be the union of the generating sets $\mathcal{T}_E, \mathcal{T}_F$ and \mathcal{T}_L for $U(E), U(F)$ and L .

2.4 Commutator relations in $\Omega(d, q)$

It is difficult, algorithmically, to construct the generating sets $\mathcal{T}_E, \mathcal{T}_F$ and \mathcal{T}_L directly. *Instead, we will construct a standard opposite pair (cf. 2.2) for groups $Q(\langle e_1 \rangle)$ and $Q(\langle f_1 \rangle)$ and then use familiar commutator relations among long root elements.* The next lemma summarises these commutator relations.

Lemma 2.1 *Let $G = \Omega(d, q)$, let $r_{e_1 f_j}(\lambda), r_{e_1 e_j}(\lambda)$, for $2 \leq j \leq m$ and $\lambda \in \mathbb{F}$, be the long root elements of $Q(\langle e_1 \rangle)$ defined as linear transformations in (2.4), and let $r_{f_1 e_j}(\lambda), r_{f_1 f_j}(\lambda)$ be the analogous long root elements in $Q(\langle f_1 \rangle)$. Then, for $2 \leq i \neq j \leq m$, we have*

1. $r_{e_1 w}(\lambda)^g = r_{(e_1 g)(wg)}(\lambda)$ for all $g \in G$.
2. $[r_{f_1 f_j}(\mu), r_{e_1 e_i}(\lambda)] = r_{e_i f_j}(\lambda \mu) \in L$.
3. $[r_{f_1 f_j}(\mu), r_{e_1 f_i}(\lambda)] = r_{f_i f_j}(\lambda \mu) \in U(F)$.
4. $[r_{f_1 e_j}(\mu), r_{e_1 e_i}(\lambda)] = r_{e_i e_j}(\lambda \mu) \in U(E)$.

2.5 Primitive prime divisors

By a fundamental theorem of Zsigmondy [Zs], if p is prime and $n \geq 2$ then there is a prime dividing $p^n - 1$ but not $p^i - 1$ for $1 \leq i < n$, except when either $p = 2, n = 6$, or $n = 2$ and p is a Mersenne prime. Such a prime is called a *primitive prime divisor* of $p^n - 1$.

We will call an integer j a $\text{ppd}^\#(p; n)$ if $j | p^n - 1$ and either $n = 2$, p is a Mersenne prime, and $4 | j$; or j is divisible by a primitive prime divisor of $p^n - 1$. We will call an element g of $\text{GL}(d, p^k)$ a $\text{ppd}^\#(p; n)$ -element if $|g|$ is a $\text{ppd}^\#(p; n)$.

Lemma 2.2 *Let $G = \Omega(d, q)$. Then the number of $\text{ppd}^\#(p; ed)$ elements in G is at least $\frac{|G|}{2d}$.*

Proof. See [KS] or [NiP].

We require a means of testing whether or not a given element of G is a $\text{ppd}^\#(p; n)$ -element. The following test is found in [NP].

Lemma 2.3 *Following a one time integer computation taking time $O(n^3 \log n \log^4 p)$, one can test whether or not any given element of G is a $\text{ppd}^\#(p; n)$ -element in time $O(\mu n \log p)$.*

3 The main algorithm

In the next two sections we present the two algorithms which together prove Theorem 1.1 when $d \geq 5$ is odd. In this section we give a Las Vegas algorithm to construct the desired generating set \mathcal{T} . We also discuss the low dimensional groups $\Omega(3, q)$ and $\Omega^+(4, q)$.

3.1 Low dimensional groups

It is well known that $\Omega(3, q) \cong \text{PSL}(2, q)$, and that $\Omega^+(4, q) \cong \text{SL}(2, q) \circ \text{SL}(2, q)$. If $G = \langle \mathcal{S} \rangle = \Omega(3, q)$, then one can find an isomorphism with $\text{PSL}(2, q)$ by specifying images in $\text{SL}(2, q)$, modulo scalars, of the elements of \mathcal{S} . Similarly, given $G = \langle \mathcal{S} \rangle = \Omega^+(4, q)$, one can find generators for each of the $\text{SL}(2, q)$ factors in the isomorphic group. In each case, we may then use our hypothesised oracle to complete the recognition of G . The complexity is dominated by the call(s) to the $\text{SL}(2, q)$ -oracle, namely $O(\chi)$.

In the proof of Theorem 1.1. for d odd, we encounter the groups $\Omega^+(4, q)$ in the following algorithmic setting. Given $G = \langle \mathcal{S} \rangle = \Omega(d, q) = \Omega(V)$ for $d \geq 5$ odd, and a subset \mathcal{U} of G such that $[V, \mathcal{U}]$ is a nonsingular 4-space of maximal Witt index, we need to recognise, constructively, when $\langle \mathcal{U} \rangle \cong \Omega^+(4, q)$. One can easily compute the action of the elements of \mathcal{U} on $[V, \mathcal{U}]$ to get the 4-dimensional representation, find the two $\text{SL}(2, q)$ factors, and then call the oracle to constructively recognise each factor. Once the isomorphism $\langle \mathcal{U} \rangle \cong \Omega^+(4, q)$ has been established,

we will also need to write an SLP from \mathcal{U} to any given element $g \in \Omega^+(4, q)$. This is done using the $\text{SL}(2, q)$ -oracle at a cost of $O(\chi)$ per element.

We may assume that whenever $\langle \mathcal{U} \rangle \cong \Omega^+(4, q)$, a call to the constructive test above will confirm this with probability at least $3/4$.

3.2 Constructing \mathcal{T}

We do this in several stages. First we construct generators for groups $Q(x)$ and $Q(y)$ corresponding to some pair x, y of singular points with $y \notin x^\perp$. We then obtain the elements of a standard opposite pair (B_x, B_y) for $Q(x)$ and $Q(y)$ using SLPs from these constructed generators. Finally, we show how to construct \mathcal{T} using SLPs from $B_x \cup B_y$.

Standard bases. By Remark (vi) following Theorem 1.1, we may assume that we have constructed a $d \times d$ matrix Φ representing a nonsingular quadratic form φ on V , preserved by G .

Let W be any subspace of V . Then we can use linear algebra to compute the perp space W^\perp of W with respect to the bilinear form associated to φ . Also, if W is a nonsingular subspace of dimension $n \leq d$, of the same isometry type as V , then it is not hard to see that there is a Las Vegas algorithm to construct a standard basis for W . This involves recursively finding perp-spaces inside W , together with $\lfloor n/2 \rfloor$ calls to the discrete log oracle. The latter computations dominate the complexity, giving a running time of $O(n\chi)$.

Constructing $Q(x)$. This part of the routine is a brief commentary on the methods of [KS] for finding $Q(x)$ in the case that we are interested in. In fact, we assume that $q \geq 17$, since we can find $Q(x)$ for smaller q , within the required time constraints, using [KS] **4.2.1, Case 1**.

Step 1. Our first task is to find generators for a naturally embedded $\Omega^+(4, q)$ subgroup of G . Choose up to $16(d-3)$ random elements of G , and use Lemma 2.3 to find one, τ , such that $|\tau|$ is divisible by a $\text{ppd}^\#(p; k(d-3))$ and by a $\text{ppd}^\#(p; k)$. Set $z \leftarrow q^{(d-3)/2} + 1$ and $\sigma \leftarrow \tau^z$. Choose up to 2^{10} random elements g of G , and for each use **3.1** and the $\text{SL}(2, q)$ -oracle to test whether the group $\langle \sigma, \sigma^g \rangle$ is isomorphic to $\Omega^+(4, q)$. If so, then set $J \leftarrow \langle \sigma, \sigma^g \rangle$. Then J is a naturally embedded $\Omega^+(4, q)$ subgroup.

Remark 3.1 *In practice, it would be inefficient to make 2^{10} calls to a constructive oracle. Instead, one could use a much faster implicit recognition algorithm for $\Omega^+(4, q)$, and run the constructive algorithm only when we are fairly sure we have the group we require.*

Correctness. Each element τ of order divisible by a $\text{ppd}^\#(p; k)$ and by a $\text{ppd}^\#(p; k(d-3))$ must split V as $V = V_1 \perp V_2^+ \perp V_{d-3}^-$ of V , inducing an element of order dividing $q-1$ on V_2^+ and of order dividing $z = q^{\frac{d-3}{2}} + 1$ on V_{d-3} . Since $(q-1, z) = 2$, $\sigma = \tau^z$ induces an element of

ppd[#]($p; e$)-order on V_2^+ and 1 on $V_2^{+\perp}$. In particular, if $J \cong \Omega^+(4, q)$, then J has 4-dimensional support $[V, J] = V_2^+ \oplus V_2^{+g}$, so J is a naturally embedded $\Omega^+(4, q)$ subgroup.

Reliability. By Lemma 2.2, there are at least $\frac{|G|}{4(d-3)}$ elements of G having the desired order, so that at least one of our random choices succeeds with probability $> 1 - e^{-4}$. By [KS], Lemma 4.12, two ppd[#]($p; k$) elements, having two dimensional support, generate a naturally embedded $\Omega^+(4, q)$ with probability at least $1/640$. For such a group, the test in **3.1** will confirm this constructively with probability at least $3/4$. Therefore, for each choice of g , with probability at least $\frac{3}{4} \times \frac{1}{640}$, we constructively recognise a group $\langle \sigma, \sigma^g \rangle \cong \Omega^+(4, q)$, so that at least one of our 2^{10} choices σ^g gives the desired group with probability $\geq 1 - (1 - 3/2560)^{2^{10}} > 1 - 1/e$.

Complexity. $O(d^3 \log d \log^4 q + \xi d + \mu d^2 \log q)$ for the $O(d)$ ppd tests (using Lemma 2.3) together with $O(\chi)$ for the 2^{10} calls to the test in **3.1**.

Step 2. Compute $V_2^+ \leftarrow [V, \sigma]$ and $V_4^+ \leftarrow \langle V_2^+, [V, \sigma^g] \rangle$. Find the two singular points x and y of V_2^+ (these are the unique points of V_2^+ fixed by σ). Use **3.1** and the $\text{SL}(2, q)$ -oracle to find a $2k$ -element (where $k = \log_p q$) generating set \mathcal{A}_J for the group $Q_J(x) = O_p(J_x)$. Compute a standard basis $\{e_1, e_2, f_1, f_2\}$ for V_4^+ such that $x = \langle e_1 \rangle$ and $y = \langle f_1 \rangle$. Use the oracle again to construct the elements $j(y) : e_i \mapsto f_i, f_i \mapsto e_i$ for $i = 1, 2$, and $a : e_1 \mapsto \rho^2 e_1, e_2 \mapsto e_2, f_1 \mapsto \rho^{-2} f_1, f_2 \mapsto f_2$ of J . Note that the oracle constructs each element of $\mathcal{A}_J \cup \{a, j(y)\}$ using an SLP from $\{\sigma, g\}$.

Complexity. $O(\chi \log q)$ to construct \mathcal{A}_J and the elements a and $j(y)$.

Step 3. We now give a Monte Carlo algorithm to construct the group $Q(x) = O_p(G_x)$: set $\mathcal{A} \leftarrow \cup_{i=0}^{d-2} \mathcal{A}_J^{\tau^i(q-1)}$ and return $\langle \mathcal{A} \rangle$.

Correctness. We need to show that with high probability, $\langle \mathcal{A} \rangle$ is the target group $Q(x)$.

Note that $\langle \mathcal{A}_J \rangle = Q_J(x)$ is a hyperbolic line of the target group $Q(x)$ and that τ acts irreducibly on a hyperplane of $Q(x)$. If $Q_J(x)$ is contained in this hyperplane, then the normal closure $\langle Q_J(x)^{\langle \tau \rangle} \rangle$ is precisely the hyperplane; otherwise it is all of $Q(x)$. The probability that the hyperbolic line lies inside a given hyperplane is at most $\frac{1}{q-1} \leq \frac{1}{16}$. If $Q_J(x)$ is not contained in the hyperplane of τ , then by an argument similar to [KS] Lemma 3.9, $\langle Q_J(x)^{\langle \tau \rangle} \rangle = \langle \mathcal{A} \rangle$. Hence, with probability at least $\frac{15}{16}$, $Q(x) = \langle \mathcal{A} \rangle$.

Complexity. $O(\mu dk)$ for the construction of \mathcal{A} .

Remark 3.2 *In the unlikely event that $\langle \mathcal{A} \rangle$ is only a hyperplane of $Q(x)$, this failure will be uncovered when we construct the opposite pair (B_x, B_y) below. Hence, we will soon upgrade the algorithm for $Q(x)$ to Las Vegas.*

Total Reliability. With probability at least $1 - (\frac{1}{e} + \frac{1}{e^4} + \frac{1}{16}) > \frac{1}{2}$, we constructed a set \mathcal{A} , and elements $j(y)$ and a , such that $Q(x) = \langle \mathcal{A} \rangle$, $Q(y) = \langle \mathcal{A}^{j(y)} \rangle$ and a acts as an element of order $(q-1)/2$ on V_2^+ , and is the identity on the perp of this space.

The remainder of the algorithm is deterministic.

Changing basis. Compute a standard basis for the $d - 4$ -space $V_4^{+\perp}$. Augment it with the one found earlier for V_4^+ to obtain a standard basis $\mathfrak{B} = \{e_1, \dots, e_m, v, f_1, \dots, f_m\}$ for V . Write each element of $\mathcal{S} \cup \mathcal{A}$, together with the elements $j(y)$ and a , relative to \mathfrak{B} .

Complexity. $O(\mu(|\mathcal{S}| + d \log q) + d\chi)$.

The opposite pair (B_x, B_y) . In the next part of the algorithm, we will construct the nice \mathbb{F}_p -bases B_x and B_y for the groups $Q(x)$ and $Q(y)$ described in section 2.2. Observe that we need only find B_x , since $B_y = B_x^{j(y)}$, which is easily seen by taking an element of B_x and conjugating with $j(y)$ using Lemma 2.1(1).

Step 0. We first describe simple, fast, procedures for computing with the isometry $\psi_x : \langle e_1, f_1 \rangle^\perp \rightarrow Q(x)$ (cf. section 2.2), thereby demonstrating that linear algebra in $Q(x)$ relative to $B_x = \psi_x(\mathfrak{B}_\rho)$ is algorithmically equivalent to linear algebra in $\langle e_1, f_1 \rangle^\perp$ relative to \mathfrak{B}_ρ .

Given any $w \in \langle e_1, f_1 \rangle^\perp$, it is easy to write down the matrix representing the linear transformation $\psi_x(w) = r_{e_1 w}(1) \in Q(x)$. Conversely, let $u \in Q(x)$. Compute $f_1 u$ and ignore coefficients of e_1 and f_1 in order to obtain a vector w in $\langle e_1, f_1 \rangle^\perp$.

Step 1. Using linear algebra in $Q(x)$, find an \mathbb{F}_p -basis for $Q(x)$ inside \mathcal{A} , and discard the remaining elements of \mathcal{A} .

Note: If \mathcal{A} generates only a hyperplane of the target group $Q(x)$, then we will uncover this failure at this point in the algorithm. Hence, we have now upgraded the algorithm for finding $Q(x)$ to Las Vegas (cf. Remark 3.2).

Step 2. We know the elements of our target standard basis B_x but we now need to *construct* them using SLPs from \mathcal{A} . Write each element of \mathcal{A} as an \mathbb{F}_p -linear combination of elements from B_x . This gives rise to a $k(d - 2) \times k(d - 2)$ base change matrix C from \mathcal{A} to B_x . Set $D \leftarrow C^{-1}$. Then the rows of D are \mathbb{F}_p -vectors of length $k(d - 2)$ giving each element of B_x as a linear combination of the \mathbb{F}_p -basis \mathcal{A} for $Q(x)$. Use these \mathbb{F}_p -vectors to write down SLPs of length $O(d \log q)$ from \mathcal{A} to each element of B_x .

Complexity. The only thing being used here is linear algebra on a $k(d - 2)$ dimensional vector space over \mathbb{F}_p . Each of the $O(kd)$ SLPs has length $O(d \log q)$, giving a total time of $O(d^2 \log^2 q)$.

From (B_x, B_y) to \mathcal{T} . It is well known that G is generated by the groups $Q(x)$ and $Q(y)$. Let \mathcal{T} be the union

$$\mathcal{T} = \mathcal{T}_L \cup \mathcal{T}_E \cup \mathcal{T}_F,$$

where $\mathcal{T}_L, \mathcal{T}_E$ and \mathcal{T}_F are respectively the generating sets for the groups $U(F), U(E)$ and L , defined in (2.6), (2.7) and (2.8) respectively. We now show how to construct the elements of these three sets using SLPs from $B_x \cup B_y$.

Constructing \mathcal{T}_L : By Lemma 2.1(2), for $0 \leq l < k$ and $2 \leq i < j \leq m$, $[r_{f_1 f_j}(\rho^l), r_{e_1 e_i}(1)] = r_{e_i f_j}(\rho^l)$, so that each root element of \mathcal{T}_L is obtained using an SLP of length 4 from $B_x \cup B_y$. *Time:* $O(d^2 \log q)$, since we need not compute each commutator inside G (cf. Remark (iv) following Theorem 1.1.)

Constructing \mathcal{T}_E and \mathcal{T}_F : We do this in three stages. First, use Lemma 2.1(3,4) to write SLPs of length 4 from $B_x \cup B_y$ to the long root elements $r_{f_i f_j}(\rho^l) \in U(F)$ and $r_{e_i e_j}(\rho^l) \in U(E)$ for $0 \leq l < k$ and $1 \leq i < j \leq m$. *Time:* $O(d^2 \log q)$.

Next, let c be any element of L such that $c : e_1 \mapsto e_2 \mapsto \dots \mapsto e_m$. Use 4.1 to write an SLP of length $O(d)$ from \mathcal{T}_L to such a c . *Time:* $O(d^2 \log q)$, since only $O(d)$ row/column operations are required.

Finally, we use c to construct the remaining short root elements of \mathcal{T}_E and \mathcal{T}_F . Since $c \in L$, it follows that for each short root element $r_{f_1 v}(\rho^l)$ ($0 \leq l < k$), $r_{f_1 v}(\rho^l)^{c^i} = r_{(f_1 c^i)(v c^i)}(\rho^l) = r_{f_{i+1} v}(\rho^l)$ for $1 \leq i < m$. Hence, we obtain SLPs of length $O(\log m)$ from $B_x \cup B_y \cup \{c\}$ to the short root elements of \mathcal{T}_E and \mathcal{T}_F . *Time:* $O(kd \log d)$.

This completes the construction of the generating sets \mathcal{T}_E and \mathcal{T}_F , and hence the construction of \mathcal{T} . The total cost of this construction is $O(d^2 \log q)$.

Remark 3.3 *The generating sets $\mathcal{T}_L, \mathcal{T}_E$ and \mathcal{T}_F are the analogues of the generating sets S, W_u and W_l used in [Ce] when $G = \text{Sp}(d, q)$. Observe, however, that the groups $U(E)$ and $U(F)$ are nonabelian here.*

Some useful elements. We conclude this section by constructing elements w_0, \dots, w_{m-1} of $U(E)$ needed in the next section. These elements are products of short root elements, and we define them recursively as follows:

$$w_{m-1} = r_{e_m v}(1), \quad \text{and} \quad w_i = w_{i+1} r_{e_{i+1} v}(1), \quad \text{for } 0 \leq i < m-1.$$

Each w_i can therefore be expressed using SLPs of length at most m from \mathcal{T} .

Total time to construct \mathcal{T} . Adding together all of the complexity calculations from the various parts of 3.2, we arrive at the timing stated in Theorem 1.1.

4 Straightline programs

In this section, we give a deterministic algorithm which will write an SLP of length $O(d^2 \log q)$ from the constructed generating set \mathcal{T} to any given element $g \in G$. This will complete the proof

of Theorem 1.1. Our approach is analogous to that used in [Ce] for $G = \mathrm{Sp}(d, q)$. We omit many technical details involving matrix relations satisfied by elements of our orthogonal group, preferring rather to give an overview of the procedure.

We may assume that a given element $g \in G$ is written relative to \mathfrak{B} . In particular, we may assume that g has the block matrix form of (2.1). Our strategy is to pre- or post-multiply g by certain elements from our key subgroups $L, U(E), U(F)$ in order to filter g down the following short chain of subgroups:

$$G > G_E > G_{E,F} = L > \{I\}.$$

We therefore begin by giving subroutines to write an SLP to any given element $g \in L$ (resp. $U(E), U(F)$) from \mathcal{T}_L (resp. $\mathcal{T}_E, \mathcal{T}_F$).

4.1 SLPs from \mathcal{T}_L

Let $g \in L$ be given. Then $g = \mathrm{diag}(A, 1, A^{-t})$ for some $A \in H < GL(m, q)$ (cf. 2.3). Use Gaussian elimination simultaneously on the upper and lower blocks to write an SLP of length $O(d^2 \log q)$ from the set of root elements of \mathcal{T}_L to an element g_1 such that $gg_1 = \iota(\lambda, 1, \dots, 1)$. Since $gg_1 \in L$, it follows that λ is a square; use the discrete log oracle to find $0 \leq n < (q-1)/2$ such that $\lambda = (\rho^2)^n$. Hence $gg_1 = a^n$. Write an SLP of length $\log q$ from $\{a\}$ to a^n , and concatenate with the inverse of the SLP for g_1 , to obtain the desired SLP from \mathcal{T}_L to g .

Complexity. $O(d^3 \log q + \chi)$ for Gaussian elimination and one call to the discrete log oracle.

4.2 SLPs from \mathcal{T}_E and \mathcal{T}_F

Since $U(E)$ is simply the transpose of $U(F)$, we will just discuss the latter. We again remark that the group $U(F)$ is nonabelian.

Let $u \in U = U(F)$ be given. Then $u = u(z, M)$ for some z, M with $M + M^t + 2z^t z = 0$ (cf. 2.3). Identify z with a vector in the t.s. m -space F and compute the \mathbb{F}_p -vector $(\lambda_i^{(l)} \mid 1 \leq i \leq m, 0 \leq l < k)$ representing z relative to the \mathbb{F}_p -basis $\{\rho^l f_i \mid 0 \leq l < k, 1 \leq i \leq m\}$ for F . This is an ordered basis, ordered lexicographically by i first, then l . Write an SLP of length $O(d \log q)$ from \mathcal{T}_F to the element

$$w = \prod_{i=1}^m \prod_{l=0}^{k-1} r_{f_i v}(\rho^l)^{\lambda_i^{(l)}},$$

where the order of the product coincides with that of the basis. Then $uw^{-1} = u(0, M') \in Z(U)$, for some skew-symmetric matrix M' (note that the matrix of w is easily computed without evaluating the SLP, and that using a different ordering for the product would still give rise to a central element, but would yield a different M'). It is now an easy matter to write an

SLP of length $O(d^2 \log q)$ from \mathcal{T}_F to uw^{-1} , since the elements of $Z(U)$ in \mathcal{T}_F have the form $u(0, E_{ij}(\rho^l) - E_{ji}(\rho^l))$, for $1 \leq i < j \leq m$, $0 \leq l < k$.

Complexity. $O(d^2 \log q + \mu)$ for finding the matrix of w and computing uw^{-1} .

4.3 The general case

Let $g \in G$ be given, as in (2.1).

Step 1. Reduce to $g \in G_E$. There are two cases to consider:

Case 1: A is nonsingular. Since g preserves $(\ , \)$, we obtain the matrix relation $AB^t + BA^t + 2z_1^t z_1 = 0$, from which it follows that

$$u = u(z_1 A^{-t}, A^{-1}B) = \begin{pmatrix} I & A^{-1}z_1^t & A^{-1}B \\ & 1 & -2z_1 A^{-t} \\ & & 1 \end{pmatrix} \in U(F).$$

Use **4.2** to write an SLP from \mathcal{T}_F to u . Then $g \leftarrow gu^{-1} \in G_E$, as required.

Case 2: $\text{rank}(A) = r < m$. We reduce to case 1. Observe that pre- and post- multiplying g by elements of \mathcal{T}_L simultaneously effects elementary row and column operations on each of the entries A, B, C, D . Using two-sided Gaussian elimination, find SLPs from \mathcal{T}_L to elements $g_1, g_2 \in L$ such that $g_1 g g_2$ has entry ‘ A ’ in (2.1) of the form $\begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix}$.

Note that we do not need to evaluate these SLPs (i.e. find the matrices g_1 and g_2) in order to write down $g_1 g g_2$; we simply perform each elementary row/column operation on g .

It is not hard to check that $g_1 g g_2 w_r$, where w_r is one of the useful elements constructed at the end of section 3, has entry ‘ A ’ in (2.1) of the form $\begin{pmatrix} I_r & 0 \\ 0 & A_{m-r} \end{pmatrix}$, where A_{m-r} is a nonsingular $(m-r) \times (m-r)$ matrix. In particular, this ‘ A ’ is now nonsingular. Set $g \leftarrow g_1 g g_2 w_r$, and use case 1.

Case 1 can be thought of as an algorithmic demonstration of the well known transitivity of $U(F)$ on the set of t.s. m -spaces opposite F (i.e., those having trivial intersection with F). More precisely, in case 1, Eg is opposite F , and we constructed $u \in U(F)$ such that $Egu = E$.

Step 2. Reduce to $g \in L$. Since $g \in G_E$, Fg is opposite E . As above, we can find an element $u' \in U(E)$, directly from the matrix of g , such that $Fgu' = F$. Use **4.2** to write an SLP from \mathcal{T}_E to u' and set $g \leftarrow gu'$.

Step 3. Completion. Since $g \in L$, we can use **4.1** to write an SLP from \mathcal{T}_L to g .

By concatenating the various SLPs obtained during the three stages of the filtering process, we obtain the desired SLP of length $O(d^2 \log q)$ from \mathcal{T} to g .

Complexity: The entire routine involves only multiplying matrices, computing the inverses of matrices, and calls to **4.1** and **4.2**: time $O(d^3 \log q + \mu + \chi) = O(d^3 \log q + \chi)$.

Together with the timing estimate for section 3, this completes the proof of Theorem 1.1.

5 Other classical groups.

We mentioned earlier that the algorithm presented here is easily modified to deal with the other classical groups. In this section, we discuss the most significant alterations, including those needed for the remaining isometry types of orthogonal groups.

Constructing $Q(x)$. In our routine to construct $Q(x)$, we first found an element τ acting irreducibly on a subspace of large dimension, while centralising an element σ having two dimensional support. In order to achieve this goal for each of $G = \Omega^\pm(2n, q), \text{SU}(2n, q), \text{SU}(2n + 1, q), \text{Sp}(2n, q)$, we use elements τ having slightly different orders than the element employed here.

We then found a naturally embedded $\Omega^+(4, q)$ subgroup J ; when $G = \text{Sp}(d, q)$ (respectively $\text{SU}(d, q)$) we instead use a naturally embedded $\text{Sp}(4, q)$ (respectively $\text{SU}(4, q)$) subgroup. Once again, the key to polynomial time efficiency lies in the treatment of the low dimensional groups $\text{Sp}(4, q)$, $\text{SU}(3, q)$ and $\text{SU}(4, q)$.

Finally, when $G = \text{Sp}(d, q)$ or $\text{SU}(d, q)$, we confront the additional problem that $Q(x)$ is usually nonabelian. However, in these cases, we always have $Z(Q) = \Phi(Q) = T$ (where $\Phi(Q)$ is the Frattini subgroup and T is the group of (x, x^\perp) -transvections), and we use the action of τ on Q/T to generate Q . In some sense, we dealt with the most difficult case in section 3.2, since there, τ did not act irreducibly on Q .

Suppose that the algorithm has reached the stage where it has constructively recognised a naturally embedded four dimensional subgroup J , and has extracted generators for the analogue Q_J of Q inside J . Then, in the cases where τ acts irreducibly on Q or on Q/T , the normal closure $\langle Q_J^{\langle \tau \rangle} \rangle$ is *guaranteed* to be all of Q ; this was only *probably* true in 3.2. Hence, the subroutine constructing Q which follows the construction of the subgroup J is deterministic in many of the remaining cases.

Constructing \mathcal{T} . Again, \mathcal{T} is comprised of generators for the analogues of the groups $L, U(E)$ and $U(F)$. The structure of these groups in each case is similar to that studied here.

There are also versions of the commutator relations which allow us to pass from point stabilisers to m -space stabilisers, where m is the Witt index. Together with generators for appropriate transvection groups (when G is $\text{Sp}(d, q)$ or $\text{SU}(d, q)$), we again use these relations to directly construct \mathcal{T} from an opposite pair (B_x, B_y) .

Straightline programs. Decomposing a given element $g \in G$ as a product of elements from the subgroups $L, U(E), U(F)$, as we did in section 4, is once again similar for the other classical groups, and is somewhat simpler for $\Omega^+(d, q)$ and $\text{Sp}(d, q)$.

References

- [BCFLS] L. Babai, G. Cooperman, L. Finkelstein, E. M. Luks and Á Seress, Fast Monte Carlo algorithms for permutation groups. *J. Comp. Syst. Sci.* 50 (1995), 296–308.
- [Bt] S.Bratus, Recognition of finite black box groups, Ph.D. Thesis, Northeastern U. 1999.
- [BrK] P. A. Brooksbank and W. M. Kantor, On Constructive Recognition of a Black Box $\text{PSL}(d, q)$, (in these Proceedings).
- [Ce] F. Celler, Matrixgruppenalgorithmen in GAP. Ph. D. thesis, RWTH Aachen 1997.
- [CeLG] F. Celler and C. R. Leedham-Green, A constructive recognition algorithm for the special linear group, pp 11-26 in: *The Atlas of Finite Groups: Ten Years On (Birmingham 1995)*, LMS Lecture Note Series **249**, CUP, 1998.
- [CoLG] M. Conder, C. R. Leedham-Green, Fast recognition of classical groups over large fields (in these Proceedings).
- [CFL] G. Cooperman, L. Finkelstein and S. Linton, Constructive recognition of a black box group isomorphic to $\text{GL}(n, 2)$, pp. 85-100 in: *Groups and Computation II, Proceedings of a DIMACS Workshop* (eds. L. Finkelstein and W. M. Kantor), AMS 1997.
- [HR] D. F. Holt and S. Rees, Testing modules for irreducibility. *J. Austral. Math. Soc. (Ser. A)* 57 (1994) 1–16.
- [Ka] W. M. Kantor, Sylow’s theorem in polynomial time. *J. Comp. Syst. Sci.* 30 (1985) 359–394.
- [KLM] W. M. Kantor, E. M. Luks and P. D. Mark, Sylow subgroups in parallel. *J. Algorithms* 31 (1999) 132–195.
- [KS] W. M. Kantor and A. Seress, Black box classical groups (to appear in *AMS Memoirs*).
- [LG] C. R. Leedham-Green, The Computational Matrix Group Project (in these Proceedings).
- [Ma] P. D. Mark, Sylow’s theorem and parallel computation. Ph. D. thesis, U. of Oregon 1993.
- [Mo] P. Morje, A nearly linear algorithm for Sylow subgroups of permutation groups, pp. 270–277 in: *Proc. Int. Symp. Symbolic and Algebraic Computation*, ACM 1995.
- [NP] P. M. Neumann and C. E. Praeger, A recognition algorithm for special linear groups. *Proc. London Math. Soc.* (3) 65 (1992), 555–603.

- [NiP] A. C. Niemeyer and C. E. Praeger, A recognition algorithm for classical groups over finite fields. Proc. London Math. Soc. (1) 77 (1998), 117-169.
- [Ta] D. E. Taylor, The geometry of the classical groups. Heldermann, Berlin 1992.
- [Zs] K. Zsigmondy, Zur Theorie der Potenzreste. Monatsh. Math. Phys. 3 (1892) 265–284.

Department of Mathematics
University of Oregon
Eugene, OR 97403
U.S.A